Multimedia Tools and Applications manuscript No. (will be inserted by the editor)

# Collaborative Recommendations with Content-based Filters for Cultural Activities via a Scalable Event **Distribution Platform**

Toon De Pessemier · Sam Coppens · Kristof Geebelen · Chris Vleugels · Stijn Bannier · Erik Mannens  $\cdot$  Kris Vanhecke  $\cdot$  Luc Martens

Received: date / Accepted: date

SMIT

Abstract Nowadays, most people have limited leisure time and the offer of (cultural) activities to spend this time is enormous. Consequently, picking the most appropriate events becomes increasingly difficult for end-users. This complexity of choice reinforces the necessity of filtering systems that assist users in finding and selecting relevant events. Whereas traditional filtering tools enable e.g. the use of keyword-based or filtered searches, innovative recommender systems draw on user ratings, preferences, and

T. De Pessemier Ghent University - IBBT INTEC - WiCa 9050 Ghent, Belgium E-mail: tdpessem@intec.ugent.be S. Coppens Ghent University - IBBT ELIS - Multimedia Lab 9050 Ghent, Belgium E-mail: sam.coppens@ugent.be K. Geebelen K.U. Leuven - IBBT Distrinet 3001 Leuven, Belgium E-mail: kristof.geebelen@cs.kuleuven.be C. Vleugels VUB - IBBT 1050 Brussels, Belgium E-mail: chris.vleugels@vub.ac.be S. Bannier E-mail: Stijn.Bannier@vub.ac.be Present address: of C. Vleugels E. Mannens E-mail: erik.mannens@ugent.be Present address: of S. Coppens K. Vanhecke E-mail: Kris.Vanhecke@intec.ugent.be Present address: of T. De Pessemier

L. Martens E-mail: Luc.Martens@intec.ugent.be Present address: of T. De Pessemier metadata describing the events. Existing collaborative recommendation techniques, developed for suggesting web-shop products or audio-visual content, have difficulties with sparse rating data and can not cope at all with event-specific restrictions like availability, time, and location. Moreover, aggregating, enriching, and distributing these events are additional requisites for an optimal communication channel.

In this paper, we propose a highly-scalable event recommendation platform which considers event-specific characteristics. Personal suggestions are generated by an advanced collaborative filtering algorithm, which is more robust on sparse data by extending user profiles with presumable future consumptions. The events, which are described using an RDF/OWL representation of the EventsML-G2 standard, are categorized and enriched via smart indexing and open linked data sets. This metadata model enables additional content-based filters, which consider event-specific characteristics, on the recommendation list. The integration of these different functionalities is realized by a scalable and extendable bus architecture. Finally, focus group conversations were organized with external experts, cultural mediators, and potential end-users to evaluate the event distribution platform and investigate the possible added value of recommendations for cultural participation.

**Keywords** Recommender System  $\cdot$  Personalization  $\cdot$  Event Modeling  $\cdot$  Distributing Event Information

#### 1 Introduction

Nowadays, people tend to consult dynamic and interactive information from the Internet, rather than static hard-copy press. This shift applies to the cultural scene as well: cultural organizations have detailed websites which may help to decide how to spend leisure time optimally. Moreover, the role of these cultural websites is evolving from merely information provider to a guide that puts emphasis and structure on the supply and demand of cultural events and their accompanying assets. To deal with the vast and complex structure of cultural activities and leisure events, we believe that an event model is necessary as the fundamental basis in organizing and accessing cultural activities information. As a solution to the overload of information, a recommender system based on this event model is essential to assist users in finding relevant activities.

In this paper, we discuss the end-to-end event distribution platform CUPID<sup>1</sup> that is implemented for the Flemish cultural scene using multiple aggregated and enriched information sources. In view of its further development and refinement, a thorough evaluation of this event distribution platform and the coupled recommendation mechanism was crucial. Moreover, we elaborate on the setup and results of an interdisciplinary study aimed at evaluating the benefit and necessity of the proposed event distribution system. Preliminary results of this work have already been published [29]. However, the present paper is a rather extended version of the former and addresses also the aspects of event model linking, the recommendation algorithm, and user assessment.

The topic of information overload and personal suggestions in the context of events has been investigated in various studies. The recommendation application proposed by Kayaalp et al. [23] collects event data from several related websites, either by using web services or web scraping. This application tries to recommend events to end-users based on a combination of content-based and collaborative filtering techniques, as in

https://projects.ibbt.be/cupid

our platform. However, our recommendation platform is able to enrich the event information using the proposed event model. This way, the personal suggestions delivered by our platform contain the event descriptions, extended with additional information originating from external knowledge sources. Klamma et al. [24] have proposed a recommendation tool based on collaborative filtering for suggesting academic events like conferences, workshops, international symposiums, etc. In this research, scientific communities' analysis and visualization have been performed to provide an insight into the communities of event series. Nevertheless, this research is limited to academic events and does not handle cultural events or leisure activities. iCITY [8] is a project aimed to be a social adaptive mobile guide that exploits the semantic web technologies in a web 2.0 vision in order to provide cultural events of the city of Torino. This research is focused on improving the navigation and information retrieval and providing accurate and trustworthy information by means of social networks and users' collective knowledge. Since the main contribution of this research is in the context of improving navigation and information provisioning on the mobile platform, the topic of personal event recommendations was only briefly investigated and discussed, unlike the recommendation section in this paper. Additionally, our research provides an evaluation in terms of prediction accuracy for the personal recommendations of the event distribution platform.

The remainder of this paper is organized as follows: Section 2 describes the EventsML-G2 [20] standard completed with its conceptualization in an OWL [31] ontology as a unifying metadata model. In Section 3 we present the workflow-based bus-infrastructure that is employed to integrate the different functionalities to a continuous flow of cultural events. Section 4 further elaborates on aggregating the flow of events and the enrichment based on knowledge available in large linked data sets. These event data together with personal preferences are used by the recommendation algorithm, which is described in Section 5, to generate the best-fit cultural suggestions. User profiling and *Facebook* as an initial distribution platform are discussed in Section 6. In order to evaluate the proposed event distribution platform and reflect on the possible added value of recommendations for cultural participation, different series of focus group conversations were organized with external experts, cultural mediators, and potential end-users, as described in Section 7. Finally, we offer a brief conclusion on our research results in Section 8.

# 2 Event Modeling

### 2.1 The Ontology

The events considered in this paper are socio-cultural events offered by cultural institutions: theatre performances, concerts, festivals, exhibitions, workshops, movie releases, sport events, etc. In Flanders, these events are aggregated by CultuurNet<sup>2</sup>. CultuurNet has a running platform which allows event organizers to upload their events using a proprietary XML schema<sup>3</sup>. Subsequently, CultuurNet disseminates the events in an *ad hoc* manner. To obtain a standardized procedure for this dissemination of events and to enable content-based recommendation techniques, we developed a semantic metadata

<sup>2</sup> http://www.cultuurnet.be

<sup>3</sup> http://www.cultuurdatabank.com/XMLSchema/CdbXSD/3.0/FINAL/CdbXSD.xsd

model for storing and exchanging event information. This way, all events are stored and described in a common manner: the mapping of the event descriptions to this semantic metadata model unifies all the aggregated events. Moreover, by using a semantic metadata model, the event information can be enriched with other information sources. This allows offering details about the event, combined with e.g. information about the hotels in the neighborhood as well as information about the creator and the organizer of the event. In addition, this semantic metadata model empowers the distribution of the event information, which will be published as Linked Open Data [2]. This way, other data providers can access and interpret the data more easily. In other words, the data become more interoperable. Another advantage of using a semantic metadata model is its reasoning capability, which allows answering complex queries, e.g., filter all theatre events from a certain writer, that take place during the weekend, within a range of 15 km.

To represent the events as structured data, various standards are candidate. A popular standard for describing events is iCalendar [21], which defines an event as anything which has a certain scheduled amount of time on a calendar. This standard describes events for personal management purposes. Although, the iCalendar format is able to describe a socio-cultural event, it can not express relations between the events. For instance, a festival can consist of many smaller events, i.e. concerts or music performances, which have to be related.

The most used event model in the Linked Open Data community is the Music Ontology Events Model [9]. AudioScrobbler [28] of Last.FM<sup>4</sup> uses this model to describe events which are then linked to a user profile, for which  $FOAF^5$  is used. This way, every time a user listens to a certain song, an event is modeled. Originally, this model was intended to describe musical events, but due to its simplicity and usability, it has been proven useful in a wide range of contexts. It describes an event as anything that has a temporal and spatial dimension. Such an event is described by its participating agents, its passive factors influencing the event, its products as a consequence of the event, and a location in time as well as space. In addition, this model allows describing relations between events. However, its simplicity is also a disadvantage since the model lacks some advanced features, like pricing information, more detailed relations between events in our context.

Finally, the format that we adopted for modeling events is the EventsML-G2 [20] standard. This is a standard of the International Press Telecommunications Council<sup>6</sup> (IPTC) for conveying event information in a news industry environment. It is intended for receiving, storing and exchanging event information from organizers as well as publishing event information by news providers. This model provides the right context for our event descriptions since it allows describing events in different languages, together with their relations, the pricing information, the minimum age, etc. Typical about this standard is the use of NewsCodes<sup>7</sup>, for which IPTC is also responsible. These NewsCodes define concepts to be assigned as metadata values to news objects like text, audio- and video files, etc. This allows for a consistent coding of news metadata over the course of time in a language-agnostic way. These NewsCodes are used for categorization and have various advantages: they can easily be shared, require a definition,

<sup>4</sup> http://www.last.fm

<sup>5</sup> http://xmlns.com/foaf/spec/

<sup>6</sup> http://www.iptc.org

<sup>7</sup>http://www.iptc.org/cms/site/index.html?channel=CH0088

and allow sharing a code as well as its semantics. An example of a NewsCode, indicating the subject of the event, is *subj:01000000*, which denotes that the subject of the event is arts, culture, and entertainment. The formal definition of this NewsCode is: *Matters pertaining to the advancement and refinement of the human mind, of interests, skills, tastes and emotions.* 

Currently, the EventsML-G2 standard is described as an XML [3] schema. In order to apply semantic web techniques, we developed an OWL [31] ontology of this standard. Another possibility to make the EventsML-G2 schema usable in the Semantic Web Stack, is to describe the schema in RDFS [5]. OWL, on the other hand, is a richer language for developing models and permits to mix RDFS and OWL constructs. Because of these reasons, the EventsML-G2 semantic ontology is described using OWL. With this schema, we are able to provide a common view on the aggregated data. This common view also supports the querying of the aggregated data and the distribution of the information in a common manner. Using semantic web technologies, this semantic model also facilitates incorporating data from external data sources. As discussed in Section 2.3, this ontology is also interoperable with other popular event models, e.g., Linked Open Descripions of Events ontology [38] (LODE).

EventsML-G2 has two manners for conveying event information: as a *conceptItem* or as a *knowledgeItem*. A *conceptItem* is aimed at describing an event solely. A *knowledgeItem* is intended for bundling a set of events which are managed as a whole. Given our context, publishing and recommending events from organizers, we utilize the *conceptItem* to model the events. For interoperability issues, we modeled agents involved in an event, e.g. organizers or participants, using the FOAF ontology. This allows us to easily incorporate information about the agents from other data sets afterwards. This additional advantage is the main difference between our developed ontology and the EventsML-G2 specification.

The graphical representation of the ontology in Figure 1 shows a part of an RDF/ XML event record. The root class for conveying event information is the class *Concep*tItem, which stores some administrative information together with descriptive metadata of the event. The administrative metadata is captured by the ItemMeta class and the ContentMeta class; the descriptive metadata about the event is situated in the Concept class. The ItemMeta class stores, among other things, the publication status of the event and the provider of the event information. To express the publication status of the event, NewsCodes are used. For instance, the NewsCode stat:usable indicates the information about the event can be published. The ContentMeta class delivers details on the creation and modification time of the event information. As already mentioned, the descriptive metadata about the event is captured by the Concept class. This class stores a description of the event, the title of the event, its relations to other events, the event details, and the language of the event description. The relations to other events can be described by *owl:sameAs* for stipulating that this event is the same as another event, and by skos:broader<sup>8</sup>, skos:narrower, skos:related for pinpointing the relations to other events. The *EventDetails* class contains a more complete event description and is linked to ten other classes for storing detailed information about the event. These linked classes, representing the basic properties of an event, are:

- time: TemporalEntity: This class from the Time Ontology is able to model points in time (time:Instant) and time intervals (time:Interval). A time:Interval has relations to time:Instant through the properties time:hasBeginning and time:hasEnd. It is

<sup>8</sup> http://www.w3.org/2004/02/skos/core#

also possible to model durations by relating the *time:TemporalEntity* class to the *time:DurationsDescription* via the object property *time:hasDurationDescription*. The *EventDetails* class can refer to multiple *time:TemporalEntity* classes, because the same event can occur on the same location on different dates.

- Location: This class has properties for denoting the name of the location and the type of this resource through a NewsCode (*cpnat:poi*). This Location class is related to the POIDetails class by the object property poiDetails. This class has a twofold description of the location: one giving the contact information, which is expressed by the ContactInfo class (see hereafter), the other giving the geospatial coordinates of the location (geo:spatialThing), which are described by the Basic Geo (WGS84 lat/long) Vocabulary<sup>9</sup>.
- foaf:Agent: This class is used for describing institutions and persons. The EventDetails class is linked to this class by the object properties organizer or participant.
- Subject: Here, the subjects of the event are stored. Subjects are expressed by News-Codes and a literal description.

Other classes related to the *EventDetails* class are:

- Language: The Language class describes the languages used in the event. It stores the name of the language as well as the role of the language by an NesCode. For instance, *lrol:subtitle* indicates the language of the subtitles used in the event.
- ContactInfo: This information consists of email addresses, postal addresses, web URLs, phone numbers, and fax numbers. Besides, this class stores the role of the contact information. This role is indicated by a NewsCode. *ecirol:general* specifies the general contact information, *ecirol:ticketing* refers to the contact information for reserving tickets for the event.
- OccurStatus: The OccurStatus class stores the certainty of the occurrence of the event using a NewsCode. For instance, eostat:eos5 means that the event is planned and will certainly occur.
- Registration: This class gives extra information on the registration for the event in the form of a literal. Pricing information, for instance, is stored here.
- ParticipationRequirement: This gives information about the participation requirements for the event. The minimum age to attend the event is expressed here.
- Media: This class stores some references to certain media files which are used as descriptive metadata for the event, e.g. a promotion movie.

This ontology is published online on the website of the Multimedia Lab research group of Ghent University<sup>10</sup>. By describing all the aggregated events using this ontology, a common ground is created for exposing the event information. This common ground acts as a unifying layer, relating all the information coming from different data providers. By providing a semantic ontology for these events, content-based recommendation algorithms are able to analyze the events in detail, and additionally, semantic web techniques can be used. A good example of this is the SPARQL [35] endpoint, allowing complex queries for the events. Another benefit of a semantic ontology is that information from other data providers can easily be incorporated into the event descriptions. This data incorporation is typically used for enriching the event information.

<sup>9</sup> http://www.w3.org/2003/01/geo/wgs84\_pos#

<sup>10 .</sup> http://multimedialab.elis.ugent.be/ontologies/EventsML-G2/v1.0/EventsML.owl



Fig. 1 Graphical representation of the ontology showing its basic properties.

#### 2.2 Distribution

Once the events are enriched, they will be published as Linked Open Data (LOD), which is a style of publishing structured data on the Web, using the existing web infrastructure. LOD stipulates that first the items of interest in the domain have to be selected. Then, these resources have to be identified by HyperText Transfer Protocol<sup>11</sup> (HTTP) Uniform Resource Identifiers<sup>12</sup> (URI). This way, the user can look up these data easily. The next principle is to provide useful information for that HTTP URI. Finally, links to other data sets have to be provided, just as with HTML publishing. In practice, this means that every resource requires two representations: XHTML [34] and RDF [1]. The LOD server redirects the client to the appropriate representation by inspecting the accept header of the user's client.

The records, coming from the event providers and described in various XML formats, are mapped to our developed ontology and stored in a triple store. To accomplish this storage task, the open-source OpenLink Virtuoso triple store<sup>13</sup> is used. For publishing the records of the triple store as Linked Open Data, we rely on the open-source tool Pubby<sup>14</sup>, which is actually a Linked Data frontend for SPARQL endpoints. A SPARQL endpoint is a web service that can handle SPARQL queries, which can be considered as semantic SQL statements. The used triple store offers such a SPARQL endpoint by default. By configuring Pubby for the SPARQL endpoint, the records stored in the Virtuoso triple store are published as Linked Open Data. This means, providing HTTP URIs for all the records served by the SPARQL endpoint, providing a simple HTML interface showing the data available about each resource, and taking care of the redirects and content negotiation. This way, access points for the event information are provided interpretable for both humans and machine agents. This supports

<sup>11</sup> http://www.ietf.org/rfc/rfc2616.txt

<sup>12</sup> http://tools.ietf.org/html/rfc3986

<sup>13</sup> http://virtuoso.openlinksw.com/wiki/main/Main

<sup>14</sup> http://www4.wiwiss.fu-berlin.de/pubby/

the reuse of the data by other applications like the recommendation system, which will use these descriptions for a content-based reasoning.

# 2.3 Event Model Linking

Our developed ontology is used for unifying and disseminating the aggregated event information. To make the event descriptions even more interoperable, our ontology can be linked to an upper ontology in future work. For this upper ontology, several existing ontologies are candidate.

DOLCE-DnS Ultralite<sup>15</sup> (DUL) is an upper ontology which is a lightweight version of the DOLCE Lite Plus library<sup>16</sup> and the Descriptions and Situations ontology<sup>17</sup>. This ontology contains about 200 classes and 130 properties. The scope of the ontology is the management of social entities, e.g., organizations, collectives, plans, norms, and information objects. Although DUL is able to describe events, these events have to be linked to an object or agent, which is not always feasible in our context.

OpenCYC<sup>18</sup> is another candidate for the linked upper ontology. OpenCYC is the open source version of the CYC<sup>19</sup> technology, the world's largest and most complete general knowledge base and common sense reasoning engine. It offers an inference engine and contains the entire CYC ontology, links to WordNet synsets<sup>20</sup>, links to the FOAF ontology<sup>21</sup>, and links to Wikipedia articles<sup>22</sup>. In comparison to DUL, OpenCYC is a very large ontology. The latest version of OpenCYC, OpenCYC 2.0, is also available as a semantic ontology<sup>23</sup>, allowing the usage in conjunction with your own data, linking your data to meaningful concepts.

The best candidate for the upper ontology is the recently developed Linking Open Descriptions of Events (LODE), which is a vocabulary for describing events as LOD. In fact, it expresses mappings between existing event ontologies. This way, this ontology can be considered as an interlingua model, which links several ontologies. The ontologies covered by LODE are ABC<sup>24</sup>, CIDOC-CRM[19], DUL and the Music Ontology Event Model[9]. By linking our developed ontology to this upper ontology, all these different event descriptions can be combined and unified, solving the interoperability issues. This would allow us to combine more easily information from other data sets and to model other types of events besides socio-cultural events (see Last.FM), enabling mixed (i.e. cross-genre) recommendations. Moreover, linking our  $ontology^{25}$ to the LODE ontology would allow other data providers to incorporate our events in their data sets. Table 1 shows the main properties of our model, aligned with the approximately equivalent properties of the LODE event model. For the actual equivalence relations, the properties are linked to the LODE properties using *skos:relatedMatch*, skos:broaderMatch and skos:narrowerMatch.

<sup>15</sup> http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS\_Ultralite

<sup>16</sup> http://www.loa-cnr.it/ontologies/DLP\_397.owl

<sup>17</sup> http://www.ontologydesignpatterns.org/ont/cdns/cDnS.owl

<sup>18</sup> http://www.cyc.com/opencyc/overview

<sup>19</sup> http://www.cyc.com/cyc/technology/whatiscyc

<sup>20</sup> http://wordnet.princeton.edu

<sup>21</sup> http://xmlns.com/foaf/spec

<sup>22</sup> http://www.wikipedia.org

<sup>23</sup> http://sw.opencyc.org/downloads/opencyc\_owl\_downloads\_v3/opencyc-latest.owl.gz

<sup>24</sup> http://metadata.net/harmony/ABC/ABC.owl

 $<sup>^{25}</sup>$  For more details on this linking, see the eventsML ontology at http://multimedialab.elis.ugent.be/ontologies/ EventsML-G2/v1.0/EventsML.owl

EventsML	LODE		
date	atTime		
location	inSpace		
	atPlace		
/	involved		
participant	involved A cent		
organiser	involveuAgent		

Table 1 Mapping between our developed ontology and the LODE ontology

# **3** General Architecture for a Personalized Event Distribution System

# 3.1 Architectural Design

The proposed architecture supports the flow of personalized, recommended, cultural activities and connects the different components of our system. Figure 2 gives an overview of the used integration layer. The infrastructure is a loosely coupled bus-architecture consisting of 6 components: aggregation, categorization, enrichment, profiling, recommendation, and distribution. All 6 will be elaborated on in the next sections, but a general overview highlighting the overall architectural flow is specified here.

From an architectural viewpoint, we distinguish 3 different views from this general architecture as illustrated in Figure 3. The first flow, the import flow, illustrates which modules are used to aggregate content and store it in an internal data store. This flow starts with the content aggregation by which cultural data is retrieved from different information sources like blogs, web sites, databases, feeds, short message service (sms), etc. Different strategies for content retrieval are possible. It can be actively searched on the web (pull) or content can be sent by partners or individual end-users (push). An example of the former is retrieving event information from an  $RSS^{26}$  feed or a web crawler that actively searches for cultural information on web sites. An example of the latter is an end-user who reports on information about an event to the system by sending a text message by phone (SMS) or by filling in an online web form. Pulling content from the web is triggered by a scheduler. On predefined times, the scheduler sends a message to the input services to check the input sources for new cultural event information. Pushing content can be done on the fly. Moreover, authentication mechanisms can be used to verify the identity of the information sources. Double detection mechanisms are required to filter out events that are already present in the internal database. This can be done by removing events with the same date, the same geographical coordinates, and the same relations to other events (e.g. relations using skos:narrower). This works only if the events are well described and the dates, locations and relations are available. A logging component is utilized to provide an audit trail that can be used to understand the activity of the integration layer and to diagnose problems. Details on aggregation of cultural information in CUPID are discussed in Section 4.1.

In the second flow, basic event information that is stored in the internal database can be used for data enrichment. Enrichment means adding extra useful information, e.g., cross-referenced multimedia, to the basic event info that helps end-users in planning their activities. First, a taxonomy builder classifies events according to themes. This classification is based on the subject field in the event ontology, which is coupled to

<sup>26</sup> http://www.rss-specifications.com/



Fig. 2 The architecture of the event distribution system

the newscodes<sup>27</sup>, as described in Section 2.1. Adding this intelligent information to the event data helps to find more accurate information during content enrichment. Based on the properties of the event, its ontology can be complemented with information from other sources. For example, an event concerning a theatre play on Shakespeare can be enriched with related literature information from libraries or footage of a former, relevant play of the same theatre company. Information on neighboring hotels can be added from a tourist service, based on the location of the event. Other useful metadata, like background information of participating actors in the event, can be retrieved from information providers like DBPedia<sup>28</sup> and added to the basic event information during the enrichment phase. Details on categorization and enrichment of cultural information can be found in Section 4.2.

Finally, the end goal is to provide the end-user with accurate information via the third flow. By means of a personal profile, the infrastructure can look for alternative recommendations in addition to the initial search results. By using profiles and a recommendation engine, we aim to provide the end-user with accurate and personalized event information, complemented with interesting recommendations, as discussed in Section 5. At the end of the third flow, the results are delivered to the customers. The distribution layer provides online publication channels (e.g. web sites) with access to the different data sources. This way, events can be searched by individuals or event information can be used and published by other organizations. In the context of CUPID, we have chosen *Facebook* as an initial distribution platform, as explained in Section 6.2.

28 http://dbpedia.org/

<sup>27</sup> http://www.iptc.org/cms/site/index.html?channel=CH0103



Fig. 3 The different phases: import flow, enrichment flow & export flow

# 3.2 Implementation & Technologies

We use a Service-Oriented Architecture<sup>29</sup> (SOA) as the underlying state of the art technology for the implementation of the integration layer. An enterprise service bus (ESB) provides the open, standards-based, connectivity infrastructure for a service-oriented architecture. A service-oriented architecture enables flexible connectivity of applications or resources by representing every application or resource as a service with a standardized interface. This allows them to exchange structured information about business events quickly and flexibly. This flexibility enables new and existing

<sup>29</sup> http://opengroup.org/projects/soa/

applications to be easily and quickly combined to address changing business needs. In the context of the integration layer, this means that the different components that compose our general architecture are modeled as separate services where the SOA infrastructure allows them to exchange data with one another as they participate in business processes. Our aim is to obtain a loose coupling between the different modules of the system with services can be very heterogeneous. The flexibility supported by this technology allows for an easy integration of new sources (i.e. aggregation sources or distribution targets) and the removal or modification of existing sources without interrupting the working system. Using a standardized technology also offers us the advantage of reusable existing middleware services. Existing solutions contain many pre-built implementations of components which can be used with a minimal effort. Existing components are available in the domain of logic and orchestration, basic interfacing, databases and data manipulation, security, logging, etc.

Orchestration between services is handled by a workflow engine. This engine is integrated in the service bus architecture and supports the execution of WS-BPEL processes. Business Process Execution Language for Web Services  $^{30}$  (WS-BPEL) provides a language for the specification of business processes and business interaction protocols. WS-BPEL is built on top of XML specifications. E.g., it uses the Web Service Description Language<sup>31</sup> (WSDL) to describe the web service interfaces that participate in a process. An executable WS-BPEL process is defined by a control flow that consists of a combination of basic and structured activities. Communication in the aggregation layer involves mainly exchanging (meta)data related to the cultural events. The events data, modeled via our events ontology, can thus syntactically easily be described in RDF/XML<sup>32</sup> as WS-BPEL (and SOA in general) supports multiple protocols including the Simple Object Access Protocol (SOAP), a protocol specification for exchanging structured information between services.

The implementation of the integration layer requires a combination of different technologies. A platform that provides development support for most of the technologies we used is the Netbeans IDE<sup>33</sup>. The NetBeans IDE is an open-source, integrated development environment written entirely in Java using the NetBeans Platform and supports development of all Java application types (Java SE, web, EJB, and mobile applications) out of the box. The ESB technology we used for this project is the Open Enterprise Service Bus<sup>34</sup>. OpenESB is an open-source project with the goal of building an Enterprise Service Bus that provides a flexible and extensible platform on which to build SOA and Application Integration solutions. It is based on open standards like Java for Business Integration<sup>35</sup> (JBI) and Java Platform Enterprise Edition<sup>36</sup> (Java EE).

We encountered several benefits in adopting a service-oriented framework approach to infrastructure development. In contrast to monolithic application silos, serviceoriented approaches consist of flexible, granular, functional components that expose service behaviors to other applications via loosely coupled standards-based interfaces. Online event-data providers consist of a dynamic mix of many different types of re-

<sup>30</sup> http://docs.oasis-open.org/wsbpel/

<sup>31</sup> http://www.w3.org/TR/2007/REC-wsdl20-20070626/

<sup>32</sup> http://www.w3.org/TR/rdf-syntax-grammar/

<sup>33</sup> http://netbeans.org/

<sup>34</sup> https://open-esb.dev.java.net/

 $<sup>35 \\ \</sup>texttt{http://jcp.org/aboutJava/communityprocess/final/jsr208/index.html}$ 

<sup>36</sup> http://java.sun.com/javaee/

sources and facilities. Also different technologies provided by different partners operate on incoming event information before it is distributed to the end-users. This multiplicity of technologies and situations demands a service-oriented approach. Aggregation, categorization, enrichment, profiling, recommendation and distribution are supported through services, which can be created and modified dynamically to suit the current needs of the CUPID architecture.

# 4 Aggregation, Categorization & Enrichment

Aggregation concerns the retrieval of primary information of cultural events from different content providers. With primary information, we mean the minimal information that is necessary to identify and locate the event. In a next phase, during content enrichment, primary information on cultural activities can be extended with more facilitating and substantive information. Examples of facilitating information are 'how to get to the event with public transport' or 'how the weather will be'. Substantive information can include background information on the artist, images, press articles, etc.

# 4.1 Aggregation

Collecting primary information is performed through a network of decentralized, semicentral and centralized importing parties. Central imports are provided by a central editorial. Semi-central imports are originating from intermediaries like cities, municipalities and industries. The biggest challenge and main focus is the aggregation of event information from decentralized parties. A lot of event information is distributed by small content providers all over the Internet. They publish their events on blogs, web sites, social networks, calendars, etc. using different formats like simple HTML<sup>37</sup>, iCalendaror RSS. The main challenge of content aggregation in the context of CUPID is to provide a framework that supports the aggregation of data from heterogeneous data sources, which use different standards to publish their events. Moreover, for some of these information sources, authentication is required as well as validation of their data.

As discussed in the previous section, this framework is a service-oriented architecture in which aggregation modules are implemented using web services that can be plugged into the integration layer. Because of the loose coupling between the services, adding a new service has no significant impact on the implementation of existing modules. An aggregation module implements an interface with an external data source. For example, an RSS module can be configured with RSS feeds to aggregate cultural events that are provided by external sources, exposing their data using this format. Because the input feeds have to be aggregated periodically, the aggregation is triggered by means of a scheduler. An administrator can set the time and the interval when new data must be pulled through the management interface, which is shown in Figure 4. Next, the aggregation services send this information in the original data format to the workflow engine. The basic architecture of this import flow is illustrated in Figure 3.

To deal with different incoming data formats, a transformation is required to convert data from their aggregated format to the internal event ontology, discussed in

<sup>37</sup>http://tools.ietf.org/html/rfc2854

14

сиріс номераде	Managem scheduling	admin tools	LOGGING	VISUALISATION	ABOUT US
	cheduling Service				
Which service do you like to schedule? Aggregation:					
<u>back</u>					

Fig. 4 A screenshot of the management interface of the scheduler

Section 2. For this purpose, we use a XSL transformations engine. XSLT<sup>38</sup> is a declarative, XML-based language used for the transformation of XML documents into other XML documents. Using OpenESB as the underlying Enterprise Service Bus technology, such an engine is readily available as a JBI service. To support the aggregation of a new data format, the administrator needs to extend the XSL style sheet with the mapping of the new format to the internal event ontology. If the engine detects a template match between the aggregated event and the XSL style sheet, it will be translated accordingly; otherwise, the data will be rejected. Finally, the workflow engine orchestrates the converted EventsML-G2 data to a web service that stores these in an RDF-triplestore.

In our initial distribution platform (i.e. the first phase of the CUPID project), we implemented two aggregation services. One is a service that pulls cultural event data from the 'CultuurDatabank', a database made available by CultuurNet. The second is a configurable web service where RSS feeds can be registered for aggregation. For this, we relied on web services which made it possible to harvest the data, while using double detection based on the location and time of the event. This data is then mapped to our developed ontology, using the XSLT service engine of the service bus. In total, 44167 event records were aggregated and mapped to our EventsML ontology for testing purposes. These EventsML descriptions of the aggregated events are then categorized and enriched in a later stage before publishing the records as linked open data. This workflow is visualized in Figure 5.

<sup>38</sup> http://www.w3.org/TR/xslt



Fig. 5 Visualization of the data aggregation, categorization and enrichment workflow.

#### 4.2 Categorization & Enrichment

As the EventsML-G2 ontology has been modeled, the conversion of the (meta)data of individual event streams into RDF according to this ontology is straightforward as was described in Section 2 and Section 4.1, e.g., converting events from CultuurNet, described by a proprietary XML schema, i.e. CDBXML<sup>39</sup>, to our EventsML descriptions. However, we advocate a further step aiming at enriching the events (meta)data semantically at intake following the linked data principle<sup>40</sup>. In our case, we apply linguistic processing on the plain text contained into the 4 W-elements (what, where, who, when) of the metadata and the main description of the events.

Before enrichment, the data need to be categorized to allow a targeted search for related information. Here comes the categorization module into play. In contrast to other indexing technologies, *i.Know's Information Forensics - Smart Indexing*<sup>41</sup>, that is used in CUPID, is based on a bottom-up approach. Thus, i.Know approaches indexation and analysis of documents in a totally different manner than other technologies. Instead of the top-down approach, retrieving only the information pre-defined in an index or taxonomy, Smart Indexing starts with the document itself and the possible information it contains. Instead of generating indexation lists by means of single keyword lists, it identifies all meaningful word groups in the non-structured information stream of events data. The benefits of this methodology are clear: a) *Efficiency:* no generation of meaningless word combinations; b) *High Precision*: all information is to be retrieved. This way, a total retrieval of all available information is realized without intensive training procedures for the analyzing program. Top-down solutions are often

<sup>39</sup> http://www.cultuurdatabank.com/XMLSchema/CdbXSD/3.0/FINAL/CdbXSD.xsd

<sup>40</sup> http://linkeddata.org

<sup>41 .</sup> http://www.iknow.be

based on intensively construed taxonomies and/or ontologies, attempting to list all available information. Non-listed words and concepts, synonyms and spelling variants disappear in the haystack. On the other hand, CUPID's Smart Indexing yields more reliable results. Our taxonomy builder thus classifies events according to common found themes which can then be linked to related info.

Besides, we use  $OpenCalais^{42}$  to extract the named entities such as persons, organizations, companies, brands, locations and other events from the metadata of the events. These named entities and their type (e.g. Person, Music Group, Music Album, Event, Location, etc.) are used for querying other data sets. The extracted locations are employed for querying *Toerisme Vlaanderen*<sup>43</sup>, *GeoNames*<sup>44</sup> and *DBpedia. Toerisme Vlaanderen* is a Flemish tourism information point, with a lot of information on Flemish cities and regions, hotels, restaurants, and events. The extracted persons and organizations are used for querying the *DBpedia* data set and *BibNet*'s data set. *BibNet*<sup>45</sup> is a network of Flemish libraries, which disseminates bibliographic information of the Flemish public libraries. Furthermore, the extracted locations, persons and organizations are used for querying the related NewsML-G2 standard) to link the extracted named entities to other audio-visual media assets. The returned links to the extra information are stored and also offered to the end-user.

The main problem with enriching records is dealing with ambiguity. GeoNames, for example, returns all cities of the world when querying its data set with a single string. This ambiguity issue can be solved by including some context information, e.g. the country. Fortunately, most events contain information about the city and the country yielding accurate recognition of the location mentioned in the description. If no country is specified and the GeoNames service returns several results, the editor chooses the right enrichment option. This ambiguity problem also comes into play for enriching persons. Here, the ambiguity can be solved by looking at the person's date of birth. If the date of birth is not at hand in the description and the DBpedia or BibNet (dissemination of all bibliographic information of Flemish public libraries on authors, singers and music bands, which are described as FOAF persons) service returns several results, the choice of the enrichment is left to the editor.

In total 44167 event records were aggregated, in our distribution platform. After the categorization stage (*i.Know* and *OpenCalais*), 17734 agents (persons, groups, institutions) and 44167 locations were detected. The detected persons were enriched with information from *DBpedia* and *BibNet*. *DBpedia* had information on 8591 persons (or 48%) and *BibNet* on 7495 persons (or 42%). From the 44167 locations, *Toerisme Vlaanderen* had information on a hotel at maximum 3 kilometers distance to the event and with minimum three stars for 35140 locations (or in 80% of all the locations). For 8 records, the editors had to choose the enrichment, because of an unsolved ambiguity. This was due to a person (Luc De Vos), without information on his date of birth. All resources detected by the categorization stage were also used for enriching the event information with media assets coming from *VRT*. This resulted in linking 31415 media objects to our event descriptions, or in other words, in 50% of all detected resources.

45 http://www.bibnet.be

<sup>42</sup> http://www.opencalais.com/

<sup>43</sup> http://www.visitflanders.co.uk/

<sup>44</sup> http://www.geonames.org

<sup>46~</sup> VRT, is a publicly-funded broadcaster of radio and television in Flanders, <code>http://www.vrt.be</code>

By enriching the event information, the end-users get a lot of extra information to dig in. Together with the recommendations, this will help them to evaluate an event and to pick out the right event according to their personal preferences. Another advantage is that the event information gets semantically linked to other data sets. This incorporates the event information into the Web of Data, following the linked data principles.

### **5** Recommendation Algorithm

#### 5.1 Traditional Collaborative Filtering

New communication models for event information (e.g. blogs, social networks, etc.) stimulate the announcement and propagation of events to the general public. A big mass of people who could never have been reached in the past get acquainted to an overwhelming bulk of (cultural) activities. The offer of available events is getting more detailed, more extensive, and by the introduction of niche events, also more varied. Although this exploding offer can be seen as a way to meet the specific demands and expectations of users, it has complicated the event selection process to the extent that users are overloaded with information and risk to 'get lost': although an abundance of information is available, obtaining useful and relevant information is often a difficult task. Conventional filtering tools, like keyword-based or filtered searches, are not competent to weed out irrelevant event information. An additional filtering based on the common popularity (expressed by user ratings or consumption patterns) can assist, but requires a broad basis of user feedback before it can make reasonable suggestions. Moreover, this technique does not consider personal preferences and individual consumption behavior, since only the best known events will be pushed forward.

The overabundance of information and the related difficulty to discover the most interesting content have already been addressed in several contexts [12,25]. Online shops, like Amazon<sup>47</sup>, internet radios, like Last.FM<sup>48</sup>, and video sharing website, like You-Tube<sup>49</sup> apply recommendation techniques [12] to personalize their website according to the needs of each user. Purchasing, clicking and rating behavior are valuable information channels for online retailers and content providers to investigate consumers' interests and generate personalized recommendations [22]. Most commonly-used recommendation algorithms are based on Collaborative Filtering (CF) techniques because they generally provide better results than Content-Based (CB) techniques and require no metadata of the content [14]. To describe these recommendation systems, the 'item' concept is introduced as a general term for any kind of content or information (e.g. a book, video, or event) and accordingly, a 'consumption' is a more general expression for user feedback (like a rating or purchase) on an item.

Most literature reviews distinguish two important classes of CF, user-based and item-based, supplemented with several optional variations on them. To generate personal recommendations for a target user, user-based CF algorithms start by finding a set of neighboring users whose purchased or rated items overlap this target user's purchased or rated items. Users can be represented as an N-dimensional vector of potential consumptions, where N is the number of distinct catalogue items (i.e. the number of

<sup>47</sup> http://www.amazon.com/

<sup>48</sup> http://www.last.fm/

<sup>49</sup> http://www.YouTube.com/

available events in our system). Consumptions of items (like purchases or ratings) are recorded in the corresponding components of this vector. However, this profile vector may remain extremely sparse (i.e. containing a lot of missing values) for the majority of users who purchased or rated only a very small fraction of the available catalogue items. Next, neighborhoods of like-minded users are composed based on user similarity values. The similarity of two users, j and k, symbolized by their consumption vectors,  $U_j$  and  $U_k$ , can be measured in different ways. The most common method is to measure the cosine of the angle between the two consumption vectors [39].

$$Similarity(\mathbf{U}_j, \mathbf{U}_k) = cosine(\mathbf{U}_j, \mathbf{U}_k) = \frac{\mathbf{U}_j \cdot \mathbf{U}_k}{||\mathbf{U}_j|| ||\mathbf{U}_k||}$$
(1)

After calculating these user similarities, the algorithm aggregates the consumed items from the most similar users, eliminates the items that the target user has already purchased or rated, and recommends the remaining items to that user [27]. An alternative to this user-based CF technique is item-based CF, a technique which compares each of the user's purchased or rated items to related items and then combines the most similar items into a recommendation list. The measurement of item similarities may be performed with the same metrics as utilized to compare user profiles with user-based CF. This item-based technique is often used to calculate recommendations for big online shops, like Amazon, where the number of users is much higher than the number of items [27].

Despite the popularity of CF, its applicability is limited due to the sparsity problem, which refers to the situation that consumption data in the profile vectors are lacking or insufficient to calculate reliable recommendations. Particularly event recommendation systems suffer from sparse data sets, since average users only attend a small fraction of all the available events. As a direct consequence of this sparsity problem, the number of similar users, i.e. the neighbors of the target user, may be very limited with a user-based CF technique. Indeed, to determine the similarity, most metrics rely on the profile overlap, which might be very incomplete or even nonexistent. Furthermore, because of this sparsity, the majority of these neighbors will also have a small number of consumed items in their profile vector. Because the candidate recommendations are limited to this set of consumptions of neighbors, the variety, quality, and quantity of the final recommendation list might be inadequate. A comparable reasoning is applicable to item-based CF techniques that work on sparse profile data. Users might have consumed a small number of items, which in turn also have a limited number of neighboring items. Again, the CF algorithm is restricted to a narrow set of items to generate the personal suggestions, which is pernicious for the efficiency of the recommender.

In an attempt to provide high-quality recommendations, even based on sparse data profiles, various solutions are proposed in literature [41]. Most of these techniques utilize trust inferences, transitive associations between users that are based on an underlying social network, to deal with the sparsity and the cold-start problems [40]. Nonetheless, these underlying social networks are in many cases not available for the recommendation algorithm, insufficiently developed, or even nonexistent for (new) web-based applications that desire to offer personalized content recommendations. 'Default voting' is an extension to the traditional CF technique which tries to solve this sparsity problem without exploiting a social network. A default value is assumed as 'vote' for items without an explicit rating or purchase [4]. Although this technique enlarges the profile overlap, it can not identify more meaningful neighbors than the traditional CF approach. Other approaches to deal with sparse data profiles, such as link analysis techniques [18] or spreading activation algorithms [16] are too computationally intensive to be applied on large data sets.

An additional problem arises from the inherent nature of (cultural) activities: most events have a limited availability in time and are bound to a specific location. However, traditional CF systems can not cope at all with these time-specific items, which typically receive user feedback (i.e. ratings, reviews, comments, etc.) after they have finished [10]. So, for the majority of events that exist as a one-time occurrence, recommendations based on ratings or reviews are already meaningless as soon as they can be calculated. This time dependency of one-and-only items, like events, has been tackled in related research. Cornelis et al. developed a framework that uses fuzzy logic, allowing to reflect the uncertain information in the domain and considering the time factors of events [11]. However, the computational complexity of such a solution is high, making it difficult to employ on large-scale event distribution systems. 'Pittcult' is a cultural event recommender which provides a way for users to rate the trustworthiness of other users. Problems of existing CF systems, such as data sparsity and users with malicious intentions, are solved by the trust-based recommendation [26]. Nevertheless, in many systems these trust relationships between end-users are not available.

Consequently, generating event recommendations is a special case in which user feedback might be necessary before the actual consumption (i.e. attending the event) takes place. Therefore, end-users should be able to indicate their appreciation for an upcoming event or their intention to attend it. In our event distribution system, users can provide feedback based on four options: 'Attending the event, I like this kind of events', 'Not attending, but I like this kind of events', 'Not attending, do not recommend me similar events' and 'Attending this event, but do not recommend me similar events', as can be seen in the screenshot of the user interface (Figure 10). The last option might be interesting for cases in which the user has other motives, besides personal interests, for attending the event. E.g., parents who accompany their kids to a children's show might not want to receive recommendations for new children's shows. This rating mechanism for upcoming events will help the algorithm to discover similarities between past and future events.

Moreover because metadata is not taken into account, CF algorithms may accidentally generate recommendations for events which are sold out, already finished, or located too far from the hometown of the user. Content-Based (CB) algorithms [7], which recommend items similar to those the user liked in the past, consider this meta information of items and can partly make up for that drawback, but the collaborative feature is totally lost on them. This situation confirms the need of a hybrid CF-CB algorithm that combines community knowledge with metadata containing event specific characteristics.

### 5.2 Probability-based Extended Profile Filtering

As a response to the sparsity problem, we developed an advanced CF algorithm that extends sparse profiles with consumptions that are likely to happen in the future. These profile extensions increase the probability of profile overlap, and accordingly, the number of neighbours. Next, the additional neighbors may lead to more precise and varied content recommendations, compared to a traditional CF algorithm. Our developed event recommendation algorithm, a robust CF technique extended with CB features, consists of four successive phases. A schematic overview of this algorithm, which is called 'Probability-based Extended Profile Filtering', is provided via the flowchart in Figure 6. In the first phase, a traditional CF algorithm is employed to generate a top-N recommendation list with a corresponding confidence value for each recommendation based on the existing profiles [36]. These recommendations, which might be inadequate due to the sparsity, are not used as the final suggestions but only as an information source for subsequent calculations.

In the second phase, all the initial profiles that do not contain a minimum number of consumptions are extended. To make these sparse profiles denser, presumably future consumptions are inserted into the profile vectors. These additional consumptions are based on two information sources: the general probability and the profile-based probability that the item will be consumed in the near future. In user-based CF systems, existing user profiles are supplemented with the items that have the highest probability to be consumed by the user in the near future. The general probability that a specific item will be consumed by a specific user without a priori knowledge of that user is proportional to the current popularity of the item. So, this probability is estimated by a linear function of the popularity of the item:  $f(x) = a \cdot x + b$ , in which f(x) is the probability of consumption, x is the popularity of the item and a and b are parameters that can be optimized according to the content type. Especially for event systems, the popularity of these 'top items' can vary rapidly in time. In addition, the probability that the item will be consumed by the user can also be calculated based on the user's profile as a priori knowledge. This probability is inversely proportional to the index of the item in the personal top-N recommendation list, and can be estimated by the confidence value which is calculated by the traditional CF system in phase 1. After all, this top-N recommendation list is a prediction of the items which the user will like or consume in the near future.

In the item-based case, item profiles, which contain the users who consumed the item in the past, are supplemented with the most likely future consumers. The general probability that a specific user will consume a specific item, without any knowledge of the item, is proportional to the present intensity of the consumption behavior of that user. Again, we can use a linear function to estimate this probability. With additional a priori knowledge of that item, the calculations can be repeated. Then, the probability is inversely proportional to the index of the user in the top-N list of users who are the most likely to consume the item in the future. This list and the associated confidence values can be generated based on the results of the traditional item-based CF algorithm [36].

Based on this calculated general and profile-based probability, the user or item profiles are completed, if possible, until the minimum profile threshold is reached. Therefore, the predicted future consumptions resulting from the two methods (without and with a priori knowledge) are merged via an aggregator function. (In the benchmarks we used the maximum as an aggregator operator.) Subsequently, these predicted consumptions are marked as uncertain in contrast to the initial assured consumptions. For example, in a ticketing service for events, the actual ticket purchases will correspond to a value of 1, which refers to a 100% guaranteed consumption, whereas the potential future purchases are represented by a decimal value between 0 and 1, according to the probability value, in the profile vector. These first two phases may consist of several successive iterations until a minimum threshold for the profile size is reached.

Based on these extended profile vectors, the similarities are recalculated with the chosen similarity metric, e.g. the cosine similarity (equation 1), in a third phase. Because of the additional future consumptions, the profile overlap and accordingly the



Fig. 6 Flowchart of the recommendation algorithm

number of neighbors will be increased, compared to the first phase. For the item-based technique, these similarities can be used as an extended 'related item' section. To produce personal suggestions, a recommendation vector is generated based on these extended profile vectors. For a user-based algorithm, the recommendation vector,  $R_j$ , for user j can be calculated as:

$$\mathbf{R}_{j} = \frac{\sum_{k=1, k \neq j}^{M} \mathbf{U}_{k} \cdot Similarity(\mathbf{U}_{j}, \mathbf{U}_{k})}{\sum_{k=1, k \neq j}^{M} Similarity(\mathbf{U}_{j}, \mathbf{U}_{k})}$$
(2)

where M stands for the number of users in the system.  $U_j$  and  $U_k$  represent the extended consumption vectors of users j and k respectively, which might contain real values. Subsequently, the top-N recommendations are obtained by selecting the indices of the components with the highest values from the recommendation vector,  $R_j$ , and eliminating the items which are already consumed by user j in the past.

Finally, various contextual post-filters which take into account personal event selection criteria, re-evaluate the personal recommendations in the fourth phase of the recommendation algorithm. These filters operate on the confidence values of the candidate recommendations and remove or penalize the suggestions which do not cope with the personal selection criteria. Events that do not satisfy at all, e.g. events that are located too far from the hometown of the user, are removed from the suggestion list, since they are meaningless for the end-user. The confidence value of these events is adjusted to the lowest possible value to express the inappropriateness of the suggestion. Accordingly, suggestions which completely fulfill all the personal requirements are favored during the generation of the recommendation list. In this case, the confidence value is increased by a linear function, by which the event might receive a higher place in the top-N recommendation list. The personal selection criteria, which can be speci-

Ticketing	User A	User B	User C	User D	User E
Event 1	1				
Event 2				1	1
Event 3		1	1		
Event 4	1				
Event 5	1	1		1	
Event 6		1	1		
Event 7					1
Event 8		1	1		

Table 2 The consumption matrix of the example. Tickets purchased by the end-user are indicated by a 'one'.

fied by the end-user, are typically related to the price, the location, the language, the participation requirements, or the date of the event. The explicitly stated event preferences are matched with the characteristics of the event, which are available through the proposed event ontology, as discussed in Section 2, and the enrichment process, as explained in Section 4.2.

The following concrete example of a small event information system might give more insights in the proposed recommendation algorithm. Table 2 illustrates the purchased tickets of 5 users and 8 events in the system via the consumption matrix. The purchased tickets are indicated by a 'one' in the column of the user. E.g. user A has bought tickets for event 1, 4 and 5 in this example. Personal recommendations for the end-users can be calculated based on the data in this matrix. By way of illustration, the recommendations for user C will be calculated in this example. Therefore, a traditional user-based CF algorithm might be used. First, the neighbors of user C in the system have to be calculated based on a similarity metric (like the cosine similarity) as illustrated in Table 3. Since all standard similarity metrics are based on the profile overlap between a couple of users, only 1 neighbor of user C can be identified, namely user B. Next, the personal recommendations are discovered by selecting the events that neighbors 'consumed' (i.e. bought) in the past. Because the system will not suggest events for which user C has already bought a ticket, only 1 new event can be recommended to user C, namely event 5.

In contrast, our proposed algorithm will try to extend the profile vectors with potential future consumptions, before generating the recommendations. The general probability that an event will be consumed without a priori knowledge is proportional to the current popularity of the event. In this small example, event 5 is the most popular event (3 times consumed). So, this event might be added as a potential future consumption to the profile vector of user C and E (i.e. the users who have not yet bought tickets for this event). However, for simplicity of the example, this general probability of consuming an event will not be considered during the selection of potential future consumptions.

Additionally, the probability of consuming an event can be calculated based on the confidence value of the user's traditional CF recommendations. This confidence value is derived (e.g. by a linear combination) from the similarity values of the identified neighbors who consumed the recommended event. Table 4 illustrates the consumption matrix after the addition of the confidence values calculated by a simple CF recommendation. This matrix contains decimal values for the potential future consumptions making it less sparse than the former. This extended consumption matrix is the input

Similarity	User A	User B	User C	User D	User E
User A	1	0.2887		0.4082	
User B	0.2887	1	0.8660	0.3536	
User C		0.8660	1		
User D	0.4082	0.3536		1	0.5000
User E				0.5000	1

**Table 3** The similarity matrix calculated based on the original consumption matrix. Missingvalues indicate a zero similarity between two users. The cosine similarity is used as metric.

Table 4 The consumption matrix after addition of the potential future consumptions.

Ticketing	User A	User B	User C	User D	User E
Event 1	1	0.2887		0.4082	
Event 2	0.4082	0.3536		1	1
Event 3	0.2887	1	1	0.3536	
Event 4	1	0.2887		0.4082	
Event 5	1	1	0.8660	1	0.5000
Event 6	0.2887	1	1	0.3536	
Event 7				0.5000	1
Event 8	0.2887	1	1	0.3536	

 Table 5
 The similarity matrix calculated based on the extended consumption matrix. Missing values indicate a zero similarity between two users. The cosine similarity is used as metric.

Similarity	User A	User B	User C	User D	User E
User A	1	0.6758	0.4839	0.7961	0.3276
User B	0.6758	1	0.9637	0.7437	0.2747
User C	0.4839	0.9637	1	0.5785	0.1491
User D	0.7961	0.7437	0.5785	1	0.7752
User E	0.3276	0.2747	0.1491	0.7752	1

for recalculating the user similarities in the system which are illustrated in Table 5. Subsequently, these updated user similarities are used in equation 2 for calculating the final recommendations. The recommendations for user C, together with the accompanying prediction values are listed in Table 6. The prediction values show that the most interesting suggestion for user C is still event 5 (just like the CF predicted). However, additional event recommendations can be provided based on Table 6. In descending order of prediction value: event 2, 1, 4 and 7. Finally, some of these suggestions might be penalized or favored (i.e. receive a lower or higher prediction value), because of contextual post-filters.

Alternatively, in addition to the traditional CF, the general popularity of the events could be used to generate more recommendations (i.e. other suggestions, besides event 5 for user C). However, such a popular recommender can make no distinction between equally popular events (like 1, 4 and 7 which are each consumed by one user). In contrast, the probability-based extended profile filter can provide a (partial) ordering of these equally popular events (event 1 = event 4 > event 7).

Prediction	Prediction Value
$R_{c,1}$	0.4589
$R_{c,2}$	0.5820
$R_{c,4}$	0.4589
$R_{c,5}$	0.9657
$R_{c,7}$	0.2015

 Table 6
 The recommendations with the accompanying prediction values for user C, calculated with the probability-based extended profile filter

#### 5.3 Algorithm Evaluation Methodology

To estimate the effectiveness of personal recommendations, two different evaluation methods are possible. On the one hand, online evaluations measure the user interactivity (e.g. clicks, buying behavior) with the personal suggestions on a running service. Offline evaluations, on the other hand, use a test set with consumption behavior which has to be predicted based on a training set with consumption history. Although online evaluation methods are the closest to reality, we opted for an offline evaluation based on data sets because such an evaluation is fast, reproducible and commonly used in recommendation research.

Therefore, we compared the proposed recommendation algorithm with the traditional CF algorithm based on evaluation metrics which are generated by an offline analysis using a data set with consumption behavior. Unfortunately, the data sets that are commonly used to bench-mark recommendation algorithms (e.g. Netflix or Movielens) are not composed of evaluations of cultural events, but contain ratings of audio-visual items like videos. As a result, these data sets contain too few sparse profiles, by which they are not representative to bench-mark an event recommendation system. Since we do not possess a data set with consumption behavior based on events, we evaluated our algorithm (without the filters on event-specific characteristics) on a data set of PianoFiles<sup>50</sup>. PianoFiles is a user-generated content site that offers users the opportunity to exchange, browse and search for sheet music they like to play. Currently, users can manage their personal collection of sheet music on PianoFiles but they do not yet receive personal recommendations. The main consumption behavior, used to feed the recommendation algorithm, consists of the personal collections of the users. Each addition to a personal collection is used to populate the consumption matrix. The data set contains 401,593 items (sheet music), 80,683 active users and 1,553,586 distinct consumptions (individual additions of sheets to personal collections) in chronological order. Since the sparsity of this data set is realistic for an event recommendation system and because the topic of sheet music is more related to cultural activities (like music concerts) than the topic of online videos, we preferred this data set for the evaluations.

For evaluation purposes, we used 50% of the consumptions (the most recent ones) as the test set and the remaining 50% of the consumption records as the input data. In order to study the performance of the algorithm under data of different sparsity levels, we created ten different training sets by selecting the first 10%, 20%, 30%, until 100% of the input data. The recommendation algorithm used these different training sets in successive iterations to generate personal suggestions which were compared to the test set. As commonly done for the evaluation of recommendations under sparse

<sup>50</sup> http://www.pianofiles.com/

data [18], the test set was first filtered to only include consumptions that are possible to predict with the input data as a priori knowledge. A consumption of an item that is not contained in the input data or a consumption of a user without any consumption behavior in the input data is not possible to predict with CF techniques. By eliminating the users and items without a priori knowledge, the CF algorithms can be compared more precisely based on the users (and items) which have specified preferences. All users in this filtered test set were included into a set of target consumers. For each of these target consumers, the algorithm generated five ordered lists of 10, 20, 30, 40 and 50 recommendations respectively, which were compared with the test set. (Only the results for 10, 30 and 50 recommendations per user are shown in this paper, since the other results illustrate the same conclusions.) This offline evaluation methodology, in which a data set is chronologically split in training set and test set, is commonly used for evaluating recommendation algorithms [13].

One of the most-used evaluation metrics is the Root Mean Squared Error (RMSE) [37,6], which is also adopted by the official Netflix contest<sup>51</sup>. However, the Netflix contest is mainly focused on predicting accurate ratings for an entire set of items, while web-based applications (like online shops or event recommendation systems) are rather interested in providing the users with a short recommendation list of interesting items [6]. To evaluate this top-N recommendation task, i.e. a context where we are not interested in predicting user ratings with precision, but rather in giving an ordered list of N attractive items to the users, error metrics are not meaningful. Therefore, information-retrieval classification metrics, which evaluate the quality of a short list of recommendations, are the most suitable for an event recommendation system.

The most popular classification accuracy metrics are the precision and recall [6]. The precision is the ratio of the number of recommended items that match with future consumptions, and the total number of recommended items. In offline evaluations, the consumptions of the test set represent the future consumptions and the recommendations that match with these consumptions are called the *relevant recommendations*.

$$Precision = \frac{\# Relevant \ recommendations}{\# Recommendations}$$
(3)

The recall stands for the ratio of the number of relevant recommendations and the total number of future consumptions. Only the 'future consumptions' in the test set are considered as *relevant items* for the end-users in offline evaluations.

$$Recall = \frac{\# Relevant \ recommendations}{\# Relevant \ items}$$
(4)

It has been observed that precision and recall are inversely related and dependent on the length of the result list returned to the user [15]. If more items are returned, precision decreases and recall increases. Therefore, in order to understand the global quality of a recommendation system, we may combine precision and recall by means of the F1-measure.

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$
(5)

<sup>51</sup> http://www.netflixprize.com/

#### 5.4 Algorithm Evaluation Results

Because the item-based algorithms (the traditional item-based CF and the proposed item-based algorithm which extends profiles) generally achieved a very low performance on the PianoFiles data set, we did not include the results of any item-based technique in this paper. This poor performance is mainly due to the nature of the data set, which contains many more items than users. Therefore, forming item neighborhoods is actually much more difficult than forming user neighborhoods [18]. This also applies for (starting) event recommendation systems which contain a large content offer but a limited number of users. Furthermore, this ratio of users and items induces the big risk that item-based algorithms will trap users in a 'similarity hole', only giving exceptionally similar recommendations; e.g. once a user evaluated theatre company X positively, he/she would only receive recommendations for more plays of theatre company X [32].

Compared to this item-based CF, a user-based strategy achieved much better results on the PianoFiles data set. In a first evaluation, we bench-marked the standard user-based CF algorithm (UBCF), which operates on the initially existing profiles [36], against the user-based version of our probability-based extended profile filtering (UBExtended), which extends the sparse profiles before generating the actual recommendations. In this performance evaluation, the UBExtended algorithm was configured to extend sparse profiles to a target size of 6 consumptions and the cosine similarity (equation 1) was used as a measure to compare profile vectors in both algorithms.

The graphs in Figure 7(a), 7(c) and 7(e) illustrate the evaluation metrics (precision, recall and F1) for these two algorithms (UBCF and UBExtended). Due to the large content offer (401,593 items) and the sparsity of the data set, recommendation algorithms have a hard job to suggest the most appropriate content items for every user. Because of this, the absolute values of the evaluation metrics seem rather low. However precision and recall values between 1% and 10% are very common in bench-marks of recommendation algorithms [16], [17]. These graphs illustrate that the best results are obtained for iterations which are based on a large quantity of training data. As the size of the training set increases, more data about user behavior becomes available, including data about the behavior of new users for which no information was available in the first part(s) of the training set. These additional data make it possible to generate recommendations for more users, which explains the increasing recall value. Besides information of new users, supplementary training data contain consumption behavior of users who have already an initial profile. This extra information can refine the user preferences, which leads to a higher precision. However, after the profile size has reached a critical point, supplementary training data have no more additional information value, which leads to a stagnating precision value. Moreover, the recommendations for new users, which generally have a lower precision value because of a limited early profile, enhance this stagnation effect. At last, the F1 metric follows the progress of this precision and recall graph closely because of its definition. Besides these general trends, the graphs prove that the UBExtended algorithm outperforms the standard UBCF in all three evaluation metrics (precision, recall and F1) and for different sizes of the recommendation list. This improvement is especially noticeable for small training sets, which mainly consist of sparse user profiles.

To demonstrate this superiority of the UBExtended algorithm for sparse profiles, a second evaluation was performed. To examine the recommender performance on the subset of users with a sparse profile, the training set was submitted to an extra filter.



(a) 10 recommendations based on the initial (b) 10 recommendations based on the training training set set which contains only sparse profiles



(c) 30 recommendations based on the initial (d) 30 recommendations based on the training training set set which contains only sparse profiles

0,07

0,06

0,05

0,04

0,03

0,02

0,01



(e) 50 recommendations based on the initial (f) 50 recommendations based on the training training set set which contains only sparse profiles

Fig. 7 The evaluation of the UBCF and UBExtended algorithm based on the precision, recall and F1 metric

This filter removed all the users with more than x consumptions from the training set to simulate the situation of a very novel recommendation system without 'well-developed' user profiles. In accordance with the first evaluation in which we extended sparse profiles to a target size of 6 consumptions, we chose in this second analysis for a filter that removes all users with a profile that is larger than this target size (i.e. 6 consumptions). In this way, a standard UBCF that operates on a data set with only sparse profiles (profile size  $\leq$  6 consumptions), was compared with the UBExtended algorithm which extends these profiles to the target size (profile size = 6 consumptions) before generating recommendations. Given the long-tail distribution of the profile size in content delivery systems, this subset of sparse-profile consumers constitutes a considerable segment of the system users.

Since the extra filter modifies the training and test sets, the absolute values of this evaluation can not be compared with the absolute values of the first evaluation, which is based on the unfiltered data sets. However, the differences between the UBCF and UBExtended algorithm in this second evaluation, illustrated in Figure 7(b), 7(d) and 7(f), compared with the differences between these algorithms in the first evaluation (Figure 7(a), 7(c) and 7(e)), confirm that the performance improvement of the UBExtended algorithm increases for more sparse data sets. Finally, the graphs of this second evaluation show that for small training sets the precision might slightly fluctuate because of insufficient data and lots of new users.

Since the parameters of the UBExtended algorithm are not yet optimized, the performance difference between the two bench-marked algorithms might even increase considerably. The UBExtended algorithm extends sparse profiles until each profile contains a predefined number of consumptions. This target profile size is an important parameter that has to be optimized in function of the performance metrics. Although we have chosen a fixed size of 6 consumptions for the extended profiles in our evaluation, we believe this parameter might be a function of the general data set statistics, namely the overall sparsity of the data matrix, the number of items, and the number of users. Moreover, the procedure of extending the profiles, which is based on general and profile-based influences, can be fine-tuned. An optimal balance between this general and profile-based information to extend the profiles might result in more precise recommendations. Finally, some typical CF parameters have to be determined, such as the similarity metric and the number of neighbors used to calculate the recommendations.

#### 5.5 A Scalable Implementation of the Recommendation Service

The recommendation service is designed to efficiently generate personal suggestions, based on the proposed algorithm, for a potentially high number of users of the event distribution platform. To accomplish this objective, we opted for a highly scalable approach (as illustrated in Figure 8) to store and process user and item metadata completed with online-generated user consumption behavior. Moreover, due to the computational burden of most (collaborative) recommendation algorithms, the generation of personal suggestions has to be realized as a distributed computation task.

The proposed recommender service consists of two web services and three small, independent applications working together to collect user behavior and metadata content from participating web sites and to generate high quality recommendations for their visitors. The web services act as a communication channel between the recommendation system and the other parts of the event distribution platform. The *Data Insertion* web service deals with the processing of new users, events and feedback on events, which are fed into the recommender. Based on these data, personal event recommendations are generated which can be queried through the *Recommendation web service*. To ensure the scalability, these web services are behind an Amazon load balancer<sup>52</sup>, which may create more instances of the web services if the load requires this.

Each of the three applications accomplishes a single well-defined task. The first application, the *Data Inserter*, inserts users, events and consumption data into the storage system. Also for storage, Amazon Web Services are used. Dynamic data, like event metadata, which might change during time, is stored in SimpleDB<sup>53</sup>, a highly

<sup>52</sup> http://aws.amazon.com/elasticloadbalancing

<sup>53</sup> http://aws.amazon.com/simpledb



Fig. 8 The recommendation service: architectural overview

available, scalable, and flexible non-relational data store. User behavior, like event feedback and activity streams, is stored using Amazon  $S3^{54}$ , a services that can be used to store and retrieve any amount of data. Secondly, the *Calculator* pulls these data from the storage system and generates personal suggestions based on the implemented recommendation algorithm. Next, these suggestions are stored in the *Recommendation Cache*, making them immediately available for retrieval by the *Recommendation web service*. Depending on the work load, several instances of this *Data Inserter* and *Calculator* may be created. At regular intervals, the third application, i.e. the *Scheduler*, generates a calculation task which is split into disjunctive jobs for the *Calculator* instances. The results of these calculation jobs are merged and inserted into the *Recommendation Cache* as an update of the personal suggestions.

Because of performance reasons, the different applications are not in direct contact with each other, but use a queue service<sup>55</sup> for inter-process communication. Moreover, the storage system is optimized for efficiently serving two commonly-used data requests: on the one hand inserting singular user, event or feedback information originating from the *Data Insertion web service*, and on the other hand, querying massive feedback and event information to feed the instances of the *Calculator* application.

# 6 Profiling & Distribution platform

### 6.1 Profiling

The aggregated events are modeled using our developed ontology and published as Linked Open Data. The recommendation system uses this information for calculating event suggestions and filtering the recommendations based on specific event characteristics. However, the recommendation system still needs profile information from the

<sup>54</sup> http://aws.amazon.com/s3

<sup>55</sup> http://aws.amazon.com/sqs

end-users to collect personal event preferences. Nowadays, social networks are very popular. Even many event organizers and cultural institutions, e.g. *Vooruit*, have already their own social network, and hence, possess a lot of profile information. Generally, the user's profile consists of three types of information:

- static information: e.g. the user's date of birth, address, favorite book, etc. These data are used in the contextual post-filters of our algorithm to eliminate events that do not meet certain criteria (e.g. based on the age or gender of the end-user).
- dynamic information: this is information coming from the user's activity stream, e.g., what is the user listening to, what is the user's current location, etc. Since this activity stream contains the event feedback and ratings, it is used to create the consumption matrix of the CF algorithm. Other dynamic parameters, like the location of the user, are used for the content-based filters.
- the social graph: this contains all the user's connections to other users, e.g., a friend list. This information is currently not used in the recommendation algorithm but might be interesting for trust-based recommendations.

Currently, recommendation systems offer personal suggestions within the closed context of a single community, but many users have already several profiles in various social networks like *Facebook*, *Netlog* and *MySpace*. So, there is no need to create another user community with its own user profiles for this event distribution platform. The profiles a user has in different social communities can be merged in one global profile. For our case, this global profile needs to consist of static information as well as dynamic information. The dynamic information has to contain the user's current location and the personal consumptions of socio-cultural events, e.g., RSVP feedback on offered events in *Facebook*. This way, the recommendation system can calculate personal suggestions, based on this global profile, without being dependent of a certain user profile from a certain user community.

#### 6.2 Distribution

The recommendation algorithm used in the proposed event distribution platform is an advanced CF system extended with content-based features, as described in Section 5. This hybrid characteristic results in the need for event information as well as user information containing personal feedback on events. This feedback information grows as the user 'consumes' events. To exploit the collaborative features of the algorithm and calculate accurate recommendations, the recommendation service needs user information from a big community. This is the reason, why *Facebook* was chosen as initial distribution platform for the event recommendation service. Another reason for the choice of *Facebook* is that it has opened up its profiles and via a *Facebook Connect* this *Facebook* profile gets loaded into the global profile that the recommendation system uses for calculating personal recommendations. The developed *Facebook* service and *Facebook* application, as currently developed, are illustrated in Figure 9:

- The Facebook eventLogger service: the eventLogger logs all events that are maintained by an event aggregator, like CultuurNet, and transforms them into Facebook events, the event format of the adopted distribution platform. Thus all events are automatically transformed and synchronized into Facebook events, eliminating the need for cultural organizers to repeat the routine of inputting events in several distribution platforms (like Facebook) and thus also eliminating the cumbersome task of making changes to events afterwards. In the current implementation, this event-Logger receives events in EventsML-G2 format from the Flemish cultural event aggregator's database and outputs the events in *Facebook*'s native event format as documented in the *Facebook* API<sup>56</sup>. Current limitations include the inability to add images (e.g. an event poster) as this is not yet implemented in the *Facebook* API. However, this logging service could easily be extended in order to include other cultural event distribution platforms on the web (*last.fm*, *myspace*, etc.).

- The Facebook eventRecommender application: the eventRecommender offers Facebook events, generated by the eventLogger, to the end-user based on his/her personal Facebook profile. Within the application, the user can provide personal feedback on the events, which propagates back to our recommendation system to adjust the suggestions. A screenshot of the user interface is provided in Figure 10.

However, generating recommendations solely on the *Facebook* profile is not optimal. Other social networks may have additional interesting information about the user that can contribute to the profile and have a beneficial effect on the accuracy of the recommendations (e.g. the ticketing information stored by *Vooruit*). If a user buys a ticket for an event of *Vooruit* via the *Vooruit* website, this ticketing information becomes part of his/her *Vooruit* profile, and not of his/her *Facebook* profile. Therefore, we developed a global profile, which can be filled with profile information coming from different networks. By providing a *Facebook* connect on other social networks, like the *Vooruit* Website, the users from this community can be uniquely linked to their *Facebook* profile. Choosing *Facebook* as a central hub for connecting the different user profiles, is justified by the fact that *Facebook* is currently the most popular user community. This way, the recommendation system can start calculating recommendations based on this global profile, which holds information from the various profiles the user has created in different user communities.

Another distribution mechanism for the recommendation service is a widget, which event organizers can publish on their website. Through this, event institutions who do not have a user community can offer recommendations to their users. To identify the users uniquely, *Facebook Connect* is needed. Moreover, if the institution has its own user community, these user profiles can be incorporated into the global profile.

# 7 Focus Group Conversations: Opportunities of Recommendations for End-Users and Cultural Organizations

#### 7.1 Methodology and Data

In order to evaluate the CUPID model and reflect on the possible added value of recommendations for cultural participation, different series of focus group conversations were organized with external experts, cultural mediators, and potential end-users. Focus groups study people in an atmosphere more natural and relaxed than a one-to-one interview and this setting creates the possibility to explore unanticipated issues as they arise in the discussion [33]. Focus groups are useful for introducing and discussing new technologies, since the group interaction is used to 'produce data and insights that would be less accessible without the interaction found in a group' [30]. During these focus groups the research project CUPID was presented and discussed. CUPID was

<sup>56</sup> http://wiki.developers.facebook.com/index.php/API



Fig. 9 The profiling flow consists of 2 Facebook application (eventLogger and eventRecommender), the recommendation engine, CultuurNet (i.e. the event aggregator) and the website of Vooruit (i.e. an art centre in Belgium). Event information and personal recommendations are published on Facebook and other social networks, like Vooruit (1). User profiles and activity streams from these social networks are sent to the recommendation engine (2). Users can provide feedback on their recommendations (3).

presented not only as an end-to-end events distribution platform on the long-term, but also as a *Facebook* application, the short-term result of the project that was finished at the moment of the focus group conversations.

For the target group of external experts, two focus group conversations were set up. One with 'conceptual-technical experts' (7 participants) and another with 'experts with macro-perspective' (6 participants). While the first conversation brought together web developers and application developers, the latter consisted of experts in the field



Fig. 10 A Screenshot of the user interface showing personal event recommendations.

of e-culture, policy, communication and participation. The focus group conversations started with a presentation of the general CUPID model. Next, the participants were asked if the model is realistic and what the pros and cons could be. Further, we elaborated in detail on possible pitfalls, technical difficulties and necessary conditions for the model to succeed. Finally, we went into detail into the different aspects of the model, namely profile creation, distribution and control, and recommendations, as well as possible alternatives to realize these aspects were discussed.

'Cultural mediators' are actors in between the cultural product and the cultural participant. For this target group, six focus group conversations were set up, respectively with umbrella organizations (8 participants), major cultural organizations (12 participants), small cultural organizations (5 participants), media (6 participants), cities and municipalities (5 participants), and social networks and weblogs (8 participants). As introduction we asked the participants about their current policy and use of social media, profiling techniques and recommendations in their sector/organization. Next, the CUPID short and long term model were presented after which we discussed the possible added value and the willingness of a possible integration of the CUPID model in their organizations. Special attention was given to the necessary conditions and possible pitfalls.

For the target group of potential end-users six focus group conversations were set up. Contrary to the other series of conversations, the focus groups with potential endusers were not divided according to sub-groups, but were set up at four different cities in order to maximize the response. Respectively two focus groups were set up in Ghent (11 and 8 participants), one in Bruges (16 participants), two in Brussels (9 and 5 participants), and one in Hasselt (3 participants). In total 52 potential end-users participated in our research. As introduction we asked the respondents about their cultural preferences and how cultural participation is established. More specifically, we asked how information is retrieved, and what the role is of social networking sites in this process. Next, the CUPID short and long term model were presented and the respondents' first reactions were gathered. Further, we discussed into details the following topics: receiving recommendations, the integration of profiles by means of *OpenID* and *Facebook Connect*, and sharing of information and privacy. The conversations were concluded with the question if CUPID would lead to more cultural participation.

In the next subsections, some topics discussed during the focus groups will be highlighted: the search for cultural event information, experiencing information abundance, the demand for recommendations, and the role of privacy in online profiles. Furthermore, we elaborate on the view on CUPID of umbrella organizations, major cultural organizations, and small cultural organizations.

### 7.2 An Emerging Demand for Recommendations

It is often claimed that since most people have limited leisure time and the offer of cultural activities to spend this time is enormous, picking the most appropriate events becomes increasingly difficult for end-users. The participants in the focus group conversations confirmed this proposition. A lack of resources, in particular of time and budget, ensures that people can select only a certain amount of cultural events to go to. Furthermore, in the decision making process to go to certain cultural events, two elements are crucial, namely interest and friends.

# "A certain interest in a cultural discipline or event always comes first. Then you look for information about it."

As quoted above, every participation starts with having an interest in certain cultural disciplines. If people are not interested in a particular cultural discipline or event, it is unlikely that they will participate. If they are interested, the process of searching for information begins. Next to the claim that picking the most appropriate events has become increasingly difficult for end-users, it is also often claimed that people, nowadays, tend to consult the Internet, rather than hard-copy press. Our research shows indeed the Internet is increasingly becoming an important source for information about cultural events. People consult online, among others, the websites of particular cultural institutions, online agendas, social networking websites and are subscribed to newsletters. Moreover, for certain cultural activities as going to the movies, the Internet has become the most important source for information and in addition for buying tickets. Also for other cultural disciplines such as concerts, the Internet has become the most important source for information.

Nevertheless, offline sources of information can not be dismissed. For the most part, online and offline information are complementary. For example, at the start of the cultural season people gather the programmes of different cultural institutions, skim through the events and make a first selection. Next, further information is obtained online from the websites of the cultural institutions; also tickets are usually purchased online. Next to printed programmes, also posters, flyers, reviews in newspapers and magazines, or trailers on television are important sources of information. In sum, although our respondents use various online and offline channels to gather cultural information, it is clear that the role of online information has become inevitable and is expected to increase in the future.

> "Usually, if you want to go somewhere, you want other people who you know to be there."

Next to interest, invitations and recommendations from friends are an important element in the process of cultural participation. After all, going out is most of the time a social experience, in other words, an experience one shares with others. During the discussions, several respondents mentioned the importance of going to cultural events in the company of friends. More important, some participants in the focus groups mentioned the role of a particular friend as an expert in a certain cultural discipline. For instance, when a friend who is active in the field of theatre recommends a certain play, he is trusted and one is very likely to go to the particular event. Furthermore, friends play an important role when it comes to find information on smaller events. In particular information on smaller events is found online on social networking websites. With the exception of three participants in our discussions, everybody had an account on at least one social network, with Facebook as the most common. The event function of Facebook, on which people can indicate their presence on an event, is considered as user-friendly. By using the event function, a personal calendar is created. This personal calendar is visually present on someone's personal Facebook home page and on the personal event page. In addition, the event function on Facebook is used to see where friends are going to and whether or not they would also like to go.

Two questions that are important for the scope of this article arise. First, do the respondents from our research experience a kind of abundance of cultural information? Second, is there a certain need occurring for personal recommendations? Let us start with the first question. Above, we stated that the respondents use various online and offline sources to inform themselves about cultural events. Most participants are subscribed to various online newsletters, but because of the large number of newsletters received and the large amount of information in the newsletters, they are hardly read. Some participants even have a separate email account to receive these newsletters, indicating that they regard this as too much information. Others see also the many invitations on social networking websites and Facebook in particular as an abundance of information. In consequence, the newsletters and invitations are hardly looked at and because of this, people are informing themselves even less. According to our respondents, this has two consequences. First, due to an abundance of information people tend to participate less than they really want to. Second, people tend to fall back on events they already know with the result that they do not or less experience new artists, events or genres. We illustrate this experience of an abundance of information with a few quotes from the focus group discussions.

> "There is too much context information and too much marketing. If you are looking for some specific information, for instance hours or location, it is sometimes hard to find in the overload of information. That is sometimes frustrating."

> "The more cultural events, the more information, the more you are killed with all the information, the less it interests me. I almost get apathetic of the information. In consequence I participate less."

> "I am subscribed to various newsletters and feeds, so that I would not miss anything. But after all, it is too much. My mailbox is swamped. And then you just fall back on certain activities you already know and it seems like you are becoming conservative."

> "I think these newsletters are quite useful to make a selection. It is lot information of course, but I do have a separate email address for newsletters. I would not like to have all these emails between my other emails. That would be too much."

"I think there is a bit of abundance of information. Concerning theatre, I can find my way, but not for music. Finally, there is so much that you really do not know how to choose. Or, that you make the wrong choices. I would like to receive the information in a more structured way, like an overview."

The respondent of the last quote indicates she wants to receive the supply of cultural event information in a more structured way. Other respondents endorsed this question and as such expressed their openness towards recommendations. Some have already experience with recommender systems on Amazon, Last.FM, or iTunes. Their recommendations have led to additional purchases and to the learning of other writers and artists than they already knew. Some others refer to specific big cultural organizations who have their own community and who generate recommendations themselves. However, these recommendations are tied to the supply of a particular cultural organization and the respondents prefer a recommendation system that covers the whole cultural supply. In sum, some participants in our focus group conversations have positive experience with recommendation systems and this kind of personalized information has led to more cultural participation. We illustrate again with some quotes:

> "When I buy something on Amazon, they also give some suggestions. A number of times I have bought something extra that way."

> "That way (through recommendations) I already got to know many new groups on iTunes, which I otherwise never would have encountered."

> "In the cultural sector it is each for themselves. You get recommendations from Vooruit<sup>57</sup>, from Ancienne Belgique<sup>58</sup>, but it is not covering the entire supply of events what would be an added value."

Many other participants in the focus group conversations have no experience with automated recommendations or do not understand the concept so well. Did this changed after we presented the CUPID project with the *Facebook* application on the short and the model on long term? Did our respondents see an added value of our system and would they use such a system? These issues are dealt with in the next sections.

### 7.3 User's Feedback about the CUPID Model

During the focus groups, the conversation partners gave very positive reactions towards the CUPID model. They experienced it as an added value getting recommendations, being surprised with unexpected cultural events and making use of automation. Important elements of their existing online social networks are now combined with recommendations in a cultural context. Next to that, new opportunities of CUPID were discussed in the form of the option of saving attended events into an archive and the implementation of user-generated context information.

> "If the service is running, it can be very good. The quality newspapers are less and less good cultural guides. So if you can get good recommendations in this way and you are able to determine for yourself what you will do with it, then it certainly has an added value."

<sup>57</sup> Vooruit is an arts centre in Ghent, Belgium (http://vooruit.be/)

 $<sup>^{58}</sup>$  Ancienne Belgique is a concert hall in Brussels, Belgium (http://www.abconcerts.be)

With regard to the recommendations, the potential end-users attached great importance to set filters in a good and easy way to specify the amount of recommendations, to specify on which days one wants to get recommendations and to specify locations and genres in order to get good recommendations and not an abundance of recommendations. They emphasized not only new opportunities to provide additional background information, but also recommendations from person to person. Besides that, they also experienced the convenience of implicit feedback (like Last.FM) as a very nice feature, because in this way updating a profile is less time consuming.

> "I don't really like Facebook, sometimes you get that many invitations. It looks like advertising to me. Then I log out automatically. I prefer to look for myself. Okay, I want to do something. When can I do something? What will it be and where is it? So I prefer to decide when I want to get that information for myself."

As for OpenID and social network integration, benefits were expected with regards to easily logging in with an existing profile. According to the focus group participants, these benefits outweigh the small time and effort to create a new profile. Concerning the privacy aspects of CUPID, the participants all agreed that the user must be able to control all the settings. The user should define what is private or public in the cultural profile. If both profile integration possibilities, and privacy aspects are highly customizable, the participants see no problems in investing a bit more time in a new cultural profile, provided that the CUPID system will work well and the recommendations meet the expectations.

> "I think it is good like this, because users who want to use the cultural profile and want to get recommendations, also will want to make the effort to specify their own interests and settings."

Although the project assumes more culture participation among more people, most focus group participants expect more and better participation among themselves because of the recommendation system. Indeed, the recommendations make sure the users get the information on time and the recommendations create the possibility to experience unknown cultural disciplines and events. However, this case is about the networked culture participant as a user.

"My gut feeling says that this application is not one that will reach the masses, not many more people will be participating in cultural activities, but rather the people who already participated will simply participate more, I think. Better information is provided, so they will probably go more often. But I do not think CUPID will necessarily let more people attend cultural activities."

7.4 The Value of Recommendations According to Cultural Organizations

In this section we approach the recommendation system developed within the CU-PID project from the supply side. We integrate the visions on recommendations of umbrella organizations, major cultural organizations, and small cultural organizations. For cultural organizations, being part of a recommendation system would be an extra tool in their relationship with their audience. In particular, they valued the automated distribution of their cultural events on several online platforms. Because the recommendation system uses social networks, organizations could be able to reach a larger audience. "Certainly, the system has an added value for those who are already very active in social networking websites. They can be given more refined information, because the more active you are, the more information you give, the better recommendations you can get. So, in terms of service, this is of course very beautiful."

In relation to the use of social media and profiling, the focus group conversations showed important differences between and in different sectors. Smaller organizations tend to be dependent on the interest and knowledge of a single employee. As a result, most small organizations are not or very limited active in the field of online social media. Some major cultural organizations, like major concert halls and arts centers are already using profiling strategies and give already recommendations, though these recommendations are mostly editorial. While major organizations would like to integrate on the long run the recommendation system of CUPID on their website, small organizations prefer a widget that leads their visitors to recommendations on social networking websites like *Facebook*. Above all, they prefer a global and central online cultural agenda as UiTinVlaanderen<sup>59</sup> to be responsible for the recommendations. By working with a widget, no significant additional costs or other resources are expected from (small) organizations.

Cultural organizations also have some additional questions regarding the recommendation system. For instance, it should be possible to recommend also non-event activities such as: tourist attractions and monuments, books, current exhibitions, and architecture. Next, the primary requirement demanded by small organizations concerning the offering of customized information is that their cultural events start on equal terms with the cultural events of major organizations. Finally, a number of respondents asked whether it is possible to monitor the effect of recommendations on real participation. This would win more organizations to recommend the CUPID system to their audience.

> "Is it possible to monitor the recommendations in one way or another? That the recommendations actually lead to more participation? I think that it would win more organizations over, if they really see that more people are participating in their events. That would be really interesting for us."

Except for the positive response on the recommendation service, there was also a somewhat more fundamental criticism on the CUPID model. According to a participant, recommendations can lead to an impoverishment of the cultural supply because the audience can be too much guided in what they actually can see.

"I must say that I instinctively have a bit of a dislike of that approach to work with profiles of visitors. That makes me think of the loyalty cards of supermarkets. It actually leads to an impoverishment of the supply of information because you can only see what is sent to you. I feel hesitant about those profiles. Of course, we also change our media in according to different target groups. In that sense we are now building a website on which different perspectives can be given, but where you do not first have to say: This is my age and those are my last five visits to your organization."

<sup>59~</sup> UiTinVlaanderen is an online leisure agenda for Flanders and Brussels, founded by the Flemish Ministry of Culture. (http://www.uitinvlaanderen.be/).

# 8 Conclusions

This paper describes a highly-scalable event recommendation platform for cultural activities. The recommendations offer the end-users a new way to obtain personalized content in the context of cultural events, instead of looking for and following multiple cultural organizations and events. An advanced collaborative filtering algorithm, completed with content-based filters which take into account specific event characteristics (like availability, spatio-temporal aspects, etc.) is developed and bench-marked. These bench-marks, based on offline evaluations with data sets, prove the need to adapt traditional collaborative filtering techniques to the sparse profile data of event distribution systems. To allow the incorporation of these content-based filters, we proposed an OWL ontology for the EventsML-G2 standard, which is used as a metadata model and automatically categorized and enriched via smart indexing and open linked data sets. The functional components of the platform are implemented as flexible, granular services, connected through a service-oriented bus-architecture, thereby providing advantages such as scalability, high-availability and load balancing. We used Facebook as an initial distribution platform and authentication mechanism; however, the next step in our research and development is an implementation based on open standards like  $openID^{60}$ and oAuth<sup>61</sup>. The focus group conversations showed that potential end-users are receptive for recommendations: they value the possible positive impact it could have on their cultural participation. On the other hand, potential end-users stress they want to keep control over the integration and sharing of their personal cultural information. Driven by the expressed importance of the company of friends during events, we are planning to incorporate social network relations and trust between people in our next recommendation algorithm. For cultural organizations, being part of an event recommendation platform, will be an extra tool in their relationship with their audience. In particular, they value the automated distribution of their cultural events on online platforms.

Acknowledgements The research activities that have been described in this paper were funded by Ghent University, K.U. Leuven, VUB, VRT-medialab, Interdisciplinary Institute for Broadband Technology (IBBT) through the CUPID project (50% co-funded by industrial partners), the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT), the Fund for Scientific Research-Flanders (FWO-Flanders), and the European Union.

### References

- Beckett, D. (ed.): RDF/XML Syntax Specification (Revised). W3C Recommendation. World Wide Web Consortium (2004). Available at http://www.w3.org/TR/ rdf-syntax-grammar/
- Bizer, C., Heath, T., Idehen, K., Berners-Lee, T.: Linked Data on the Web. In: Proceedings of the 17th International World Wide Web Conference – LDOW Workshop, pp. 1265–1266. Beijing, China (2008)
- Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., Yergeau, F. (eds.): Extensible Markup Language (XML) 1.0 (Fourth Edition). W3C Recommendation. World Wide Web Consortium (2006). Available at http://www.w3.org/TR/2006/REC-xml-20060816/
- Breese, J., Heckerman, D., Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 43–52. Madison, USA (1998)

<sup>60</sup> http://openid.net/

<sup>61</sup> http://oauth.net/

- Brickley, D. (ed.): RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation. World Wide Web Consortium (2004). Available at http://www.w3.org/TR/ rdf-schema/
- Campochiaro, E., Casatta, R., Cremonesi, P., Turrin, R.: Do metrics make recommender algorithms? Advanced Information Networking and Applications Workshops, International Conference on 0, 648–653 (2009). DOI http://doi.ieeecomputersociety.org/10.1109/ WAINA.2009.127
- Cantador, I., Bellogín, A., Vallet, D.: Content-based recommendation in social tagging systems. In: RecSys '10: Proceedings of the fourth ACM conference on Recommender systems, pp. 237–240. ACM, New York, NY, USA (2010). DOI http://doi.acm.org/10. 1145/1864708.1864756
- Carmagnola, F., Cena, F., Console, L., Cortassa, O., Ferri, M., Gena, C., Goy, A., Parena, M., Torre, I., Toso, A., Vernero, F., Vellar, A.: icity - an adaptive social mobile guide for cultural events. In: mobile guide 06 (2006)
- 9. Centre for Digital Music University of London: The Event Ontology (2007). Available at http://purl.org/NET/c4dm/event.owl
- Cornelis, C., Guo, X., Lu, J., Zhang, G.: A Fuzzy Relational Approach to Event Recommendation. In: Proceedings of the 1st Indian International Conference on Artificial Intelligence, pp. 2231–2242. Pune, India (2005)
- Cornelis, C., Lu, J., Guo, X., Zhang, G.: One-and-only item recommendation with fuzzy logic techniques. Information Sciences 177(22), 4906 - 4921 (2007). DOI DOI:10.1016/j.ins.2007.07.001. URL http://www.sciencedirect.com/science/article/ B6V0C-4P5R62N-3/2/cde1e1d5f5a3a2e663a0f8d2b7a152bc
- Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., Gupta, S., He, Y., Lambert, M., Livingston, B., Sampath, D.: The youtube video recommendation system. In: RecSys '10: Proceedings of the fourth ACM conference on Recommender systems, pp. 293–296. ACM, New York, NY, USA (2010). DOI http://doi.acm.org/10.1145/1864708. 1864770
- Hayes, C., Massa, P., Avesani, P., Cunningham, P.: An on-line evaluation framework for recommender systems. In: In Workshop on Personalization and Recommendation in E-Commerce (Malaga. Springer Verlag (2002)
- Herlocker, J., Konstan, J., Borchers, A., Riedl, J.: An Algorithmic Framework for Performing Collaborative Filtering. In: Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 230–237. Berkeley, USA (1999)
- Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. 22(1), 5–53 (2004). DOI http://doi.acm. org/10.1145/963770.963772
- Huang, Z., Chen, H., Zeng, D.: Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. ACM Trans. Inf. Syst. 22(1), 116–142 (2004). DOI http://doi.acm.org/10.1145/963770.963775
- Huang, Z., Zeng, D., Chen, H.: A comparison of collaborative-filtering recommendation algorithms for e-commerce. IEEE Intelligent Systems 22(5), 68–78 (2007). DOI http: //dx.doi.org/10.1109/MIS.2007.80
- Huang, Z., Zeng, D., H.Chen: A link analysis approach to recommendation with sparse data. In: AMCIS 2004: Americas Conference on Information Systems. New York, NY, USA (2004)
- International Council of Museums / ICOMs International Committee for Documentation: Definition of the CIDOC Conceptual Reference Model (2009). Available at http://cidoc. ics.forth.gr/docs/cidoc\_crm\_version\_5.0.1\_Mar09.pdf
- International Press Telecommunications Council: EventsML-G2 Specification Version 1.1 (2009). Available at http://www.iptc.org/std/EventsML-G2/EventsML-G2\_1.3.zip
- 21. Internet Engineering Task Force: Internet Calendaring and Scheduling Core Object Specification – iCalendar (2009). Available at http://tools.ietf.org/html/rfc5545
- Karypis, G.: Evaluation of Item-Based Top-N Recommendation Algorithms. In: Proceedings of the 10th International Conference on Information and Knowledge Management, pp. 247–254. Atlanta, USA (2001)
- Kayaalp, M., Özyer, T., Özyer, S.T.: A collaborative and content based event recommendation system integrated with data collection scrapers and services at a social networking site. In: ASONAM '09: Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Mining, pp. 113–118. IEEE Computer Society, Washington, DC, USA (2009). DOI http://dx.doi.org/10.1109/ASONAM.2009.41

- Klamma, R., Cuong, P.M., Cao, Y.: You never walk alone: Recommending academic events based on social network analysis. In: O. Akan, P. Bellavista, J. Cao, F. Dressler, D. Ferrari, M. Gerla, H. Kobayashi, S. Palazzo, S. Sahni, X.S. Shen, M. Stan, J. Xiaohua, A. Zomaya, G. Coulson, J. Zhou (eds.) Complex Sciences, *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 4, pp. 657–670. Springer Berlin Heidelberg (2009)
- Kurapati, K., Gutta, S., Schaffer, D., Martino, J., Zimmerman, J.: A Multi-agent TV Recommender. In: Proceedings of the 5th International Conference on User Modeling – Workshop Personalization in Future TV, pp. 1–8. Sonthofen, Germany (2001)
- Lee, D.H.: Pittcult: trust-based cultural event recommender. In: RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems, pp. 311–314. ACM, New York, NY, USA (2008). DOI http://doi.acm.org/10.1145/1454008.1454060
- Linden, G., Smith, B., York, J.: Amazon.com Recommendations: Item-to-item Collaborative Filtering. IEEE Internet Computing 7(1), 76–80 (2003)
- LinkingOpenData (W3C SWEO Community Project) Centre for Digital Music: Audioscrobbler RDF Service (2007). Available at http://www.audioscrobbler.net/
- Mannens, E., Coppens, S., De Pessemier, T., Geebelen, K., Dacquin, H., Van de Walle, R.: Unifying and targeting cultural activities via events modelling and profiling. In: EiMM '09: Proceedings of the 1st ACM international workshop on Events in multimedia, pp. 33–40. ACM, New York, NY, USA (2009). DOI http://doi.acm.org/10.1145/1631024.1631033
- Marshall, C., Rossman, G.: Designing Qualitative Research. Sage Publications, London, UK (1999)
- McGuinness, D., van Harmelen, F. (eds.): OWL Web Ontology Language: Overview. W3C Recommendation. World Wide Web Consortium (2004). Available at http://www.w3.org/ TR/owl-features/
- 32. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: CHI '06: CHI '06 extended abstracts on Human factors in computing systems, pp. 1097–1101. ACM, New York, NY, USA (2006). DOI http://doi.acm.org/10.1145/1125451.1125659
- Morgan, D.: Qualitative Research Methods Series: Vol. 16. Focus Groups as Qualitative Research. Sage Publications, Newbury Park, USA (1988)
- 34. Pemberton, S. (ed.): XHTML 1.0 The Extensible HyperText Markup Language (Second Edition). W3C Recommendation. World Wide Web Consortium (2002). Available at http://www.w3.org/TR/xhtml1/
- Prud'hommeaux, E., Seaborne, A. (eds.): SPARQL Query Language for RDF. W3C Recommendation. World Wide Web Consortium (2007). Available at http://www.w3.org/TR/ rdf-sparql-query/
- 36. Segaran, T.: Programming Collective Intelligence, 1st edn. O'Reilly (2007). URL http: //proquestcombo.safaribooksonline.com/9780596529321/
- 37. Shani, G.: Tutorial on evaluating recommender systems. In: RecSys '10: Proceedings of the fourth ACM conference on Recommender systems, pp. 1–1. ACM, New York, NY, USA (2010). DOI http://doi.acm.org/10.1145/1864708.1864710
- Shaw, R., Troncy, R., Hardman, L.: LODE: Linking Open Descriptions of Events. In: Proceedings of the 4th International Asian Semantic Web Conference. Shanghai, China (2009)
- 39. Wang, J., de Vries, A.P., Reinders, M.J.T.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 501–508. ACM, New York, NY, USA (2006). DOI http://doi.acm.org/10. 1145/1148170.1148257
- Weng, J., Miao, C., Goh, A., Shen, Z., Gay, R.: Trust-based agent community for collaborative recommendation. In: Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1260–1262. Hakodate, Japan (2006)
- 41. Yildirim, H., Krishnamoorthy, M.S.: A random walk method for alleviating the sparsity problem in collaborative filtering. In: RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems, pp. 131–138. ACM, New York, NY, USA (2008). DOI http: //doi.acm.org/10.1145/1454008.1454031