

Use Case Description Template

Use Case 0 - Definition			
Name:	<i>Media retrieval service</i>	Current Version:	<i>0.1</i>
Code:	<i>MRS</i>		
Summary:	<i>Media search and retrieval is a key task in professional media production processes. It is used in a lot of real-life production workflows. This use case describes a media indexing and retrieval component that can be integrated in SOA's. It supports both automatic and manual media retrieval tasks.</i>		
Description:	<p><i>Media retrieval is a production step that is performed in many media production workflows. Examples are plentiful: news redactionists that want to retrieve some footage for use in broadcast news, quiz makers that are looking for supporting footage to illustrate questions, documentary redactionists that are looking for interesting footage for the topic discussed in their documentary, an editor that wants immediate access to all raw footage for a specific soap episode etc...</i></p> <p><i>The main task supported by this use case is media retrieval itself. These requests for media retrieval can come from a lot of sources in a media production environment, both manual and automatic requests. Moreover, in a distributed media environment, media material and metadata is present in different repositories and in this case, a media retrieval task might need to query all these different systems simultaneously and combine the results. Therefore, standardized protocols for querying repositories and retrieving results need to be available.</i></p> <p><i>In the most simple case, a user (the process initiator) constructs a search query and the search component generates a set of results matching the query. In general however, more advanced aspects of a search system are expected in a media production workflow: advanced query building, natural language input, new result visualisation strategies, iterative query refinement, result set history and storage etc... are only some of the possible extensions upon this basic model.</i></p> <p><i>In the description above, several basic components of this use case can be identified:</i></p> <ul style="list-style-type: none"> <i>- The query input component, presenting the user one or several ways to construct a query based on the users' expectations about the results. This component can be one or several text input boxes, but is certainly not restricted to this and more advanced or intuitive query building systems can be used. Queries can originate from many different systems in a media production workflow.</i> <i>- The core search component, which matches the user input query with the available metadata in the repositories and produces a set of results that match or partially match the query, possibly along with a confidence measure or score.</i> <i>- A result presentation component, allowing a user to view and/or browse the produced results. This can be as simple as a list of results ordered by rank, but new and innovative ways to present search results have started to emerge recently.</i> <i>- Result refinement services allowing the user to drill down the result set, modify the input query or selecting individual results. A straightforward solution is modifying the original query, but other methods of drilling down on results are possible as well, e.g. selecting specific facets, adding constraints, ...</i> 		

	<p>- <i>The result output component which creates a result set that in turn can be used by other services in the media production workflow.</i></p> <p><i>In order to work effectively and efficiently, this component might create several data structures as a side task to facilitate retrieval, such as an index allowing fast retrieval, an ontology facilitating query construction, reasoning and result linking, etc...</i></p>
Initiating Actor:	<i>Redactionist, other media services</i>
Supporting Actors:	<i>Producers, annotators</i>
Inputs:	<p><i>Content flows:</i></p> <ul style="list-style-type: none"> - <i>A media file for querying the repositories (e.g. query by example)</i>
	<p><i>Information flows</i></p> <ul style="list-style-type: none"> - <i>user query input</i>
	<p><i>Control flows</i></p> <ul style="list-style-type: none"> - <i>Manual check of metadata made retrievable by the component</i> - <i>Statistics of all the components in the use case, query performance indicators</i>
Outputs:	<p><i>A set of references to media items</i> <i>(optional) Confidence scores for each item in the set</i> <i>(optional) The query that produced the set</i></p>
	<p><i>Information outputs:</i> <i>Status report of updated metadata</i> <i>Statistics about the performed search operations: processing time etc...</i></p>
Pre-conditions:	<i>Metadata for the retrievable media items is available in a metadata content repository in a standardized format</i>
Post-conditions:	<i>After input of new or updated metadata, this metadata must be available for searching.</i>
	<i>The returned set of retrieved media items must (partially) match the input query.</i>
Non-functional requirements:	
Default flow:	
<p><i>Metadata input:</i></p> <ol style="list-style-type: none"> 1. <i>The process initiator calls the media retrieval metadata input service</i> 2. <i>The process initiator passes a set of metadata that has to be added, updated or deleted in the media retrieval service. This set is passed in a standardised XML format.</i> 3. <i>The service processes the list and updates its data structures</i> 	

4. *The service returns a report of the updates to the process initiator*

Media retrieval:

1. *The process initiator starts up the media retrieval service*
2. *A. Manual mode*
 - 2.1. *The system presents a query input interface*
 - 2.2. *The process initiator builds a query with the query builder*
3. *B. Automatic mode*
 - 3.1. *The process initiator passes the input query to the system using a dedicated service interface (API)*
4. *The system translates the query in its internal format, executes the query and passes it to the different distributed repositories. It generates a set of combined results that match or partially match the query. Optionally, the results are tagged with ranking information and/or confidence scores.*
5. *(optional, manual mode) The set of results is visualised in some manner and presented to the process initiator. The initiator can browse through this set of results.1*
6. *(optional, manual mode) The initiator can drill down on the search results or refine the current query with the query builder, after which step 4 is repeated.*
7. *(optional, manual mode) The initiator can manually select one or more results from the result set and remove the other ones*
8. *The system places the final result set in a standardised format and outputs it to the process initiator.*

Exception Handling:

Exceptions can occur in several steps of this use case:

- *Exception when adding metadata. The system operator is informed of this exception*
- *Exception in the execution of a query. The service initiator is informed of this exception.*

Optional Actions:

A description of supplementary (non-essential) actions applied during the course of the process.

Questions:

Any points that will need clarification in later versions of the process definition, for example.

Related Documentation:

Any relevant external references.

Use Case History:

Version	Date	Author(s)	Changes
---------	------	-----------	---------

0.1	4 jan 2012	Mike Matton	N/A
0.2	10 jan 2012	Mike matton	Updated according to comments from JRS

Figure 1: OASIS REFERENCE MODEL APPLIED TO FIMS