Evaluation

Two scenarios have been used in this work to illustrate typical use cases for the Semantic Web-enabled Multimedia Presentation System. Given a partial implementation of SWeMPs based on the concrete decisions described in the last chapter, this chapter describes setting the SWeMPs prototype up for both scenarios, their execution and the resulting evaluation of the work based on a comparison with the aims of the Standard Reference Model for IMMPS as well as the requirements specified in the initial problem statement.

This thesis has formulated a problem statement (see section 1.4.1), considered the past and contemporary work in the problem area and from this study and particularly the SRM-IMMPS we have proposed SWeMPs – a Semantic Web-enabled Multimedia Presentation System. This proposal includes a component-based architecture and conceptual model which have been implemented in a prototype[63]. Based on this prototypical implementation we now turn to evaluation.

Firstly, we ground our choice of methodology (section 6.1) for the evaluation. We also measure the effectiveness of the approach in terms of its domain independence (section 6.2). We take the two scenarios that have been given as typical use cases (section 1.3) in the light of the initial problem statement and the requirements for a system which were drawn from that (section 1.4).

We outline the groundwork for realising each scenario by defining the knowledge, content and presentation rules that are required and describing how SWeMPs will use this data for the multimedia generation process, noting particular value demonstrated in the approach. The realisation of the two scenarios is then outlined (sections 6.3 and 6.4):

- A Family Tree tool using images taken from a Family Photo Album;
- An interactive tourism video for broadcast on an IP-enabled Digital Television.

Observations from their realisation are stated and evaluated according to the chosen methodology (section 6.5). From this, we can go on to draw our conclusions (section 6.6) about the value of the approach proposed by this research.

[63] Both the service planner and multimedia modeler components are implemented in simple forms for the purposes of this “proof of concept”. Both could be further elaborated in the future.

6.1 Methodology for the evaluation

In some areas of computer science research, e.g. software engineering, evaluation methodologies are well developed and accepted [Six,1994]. This is particularly true where elements are quantifiable [Zuse,1998], e.g. by measuring the length of time taken to fulfil a task. In software development, a common methodology is qualitative [Tansley,1993], e.g. a piece of software meets a defined set of requirements. The specific requirements are normally determined prior to the software design phase and the software evaluated against the requirements analysis.

However, there are many fields within computer science where quantitative evaluation is not possible due to a lack of relevant measurable elements, and qualitative evaluation is difficult as the field does not have a specified set of requirements which defines it. Alternatively, as the field develops, a set of commonly agreed principles are identified, the “heuristics” [Polya,1971]. The evaluation of a development in the field according to those principles is known as “heuristic” evaluation. An example of a field where heuristic evaluation is the most common approach is user interface design [Nielsen,1994]. Here, a user interface is reviewed by a group of experts and each expert assesses how much the interface complies with a set of usability heuristics. The SWeMPs prototype is not yet equipped with a friendly user interface so even with a set of heuristics for a multimedia presentation system it is too early to open SWeMPs to evaluation by user groups. With further work on the prototype a later evaluation with a user group remains a possibility.

In attempting to evaluate the approach realized by SWeMPs, a heuristic methodology is not yet possible. Quantitative elements such as time taken to generate the final presentation are less relevant to the evaluation as qualitative, i.e. to demonstrate the value of a new approach. Once such value is established, future development can concentrate on “ironing out” quantitative issues such as performance (e.g. by trying different storage and reasoning solutions for the conceptual model). Hence, we have chosen a qualitative approach to the evaluation.

The SWeMPs framework has been proposed as a basis for supporting Semantic Web-enabled multimedia presentation services, termed Intelligent Information Services (IIS), benefiting from both the distributed knowledge that will be available over the Semantic Web and the communicative potential of multimedia presentations. If this framework is to demonstrate these benefits, we need examples where information is found through reasoning over knowledge acquired through the Semantic Web and presented using synchronised multimedia to effectively communicate the information to the user. These examples serve as a means for evaluation by checking if the implemented application can support the intended scenario.

In section 1.3 we introduced two scenarios, in which a set of knowledge, content and presentation rules are to be specified and used to describe in more concrete terms the expected operation of the SWeMPs system in realising those scenarios. The scenarios function as an initial demonstrator of the SWeMPs framework, as a proof of its successful operation and as an indicator of its potential. In the first scenario we can demonstrate the successful integration of the components and rulebase into the SWeMPs prototype through provided a well defined input and receiving the expected output. In the second scenario we can make a requirements-based evaluation of the prototype.

Figure 6.1 illustrates qualitative evaluation in the software engineering lifecycle with the structure of this thesis added to illustrate the place of the different stages of the software development within the presented research. We consider Chapters 1 through 3 to form the "Feasibility Study" and the "Requirements Statement" as there we have motivated our research and studied the state of the art of relevant technologies and approaches, derived requirements and proven that the current state of the art has not met all requirements. Design has taken place first at a conceptual level and then in some more detail in chapter 4. The implementation is presented in chapter 5. Now we use the evaluation of this chapter to provide "Module Validation" (do all components act correctly?). "Integration and System Test" (does the system realise the end-to-end process?) and "Evaluation and acceptance" (can the expected scenarios be realised?).

The tasks that we decide to realise in the second scenario are designed to relate to the requirements which have been defined for next generation multimedia generation systems. In Chapter 3, an analysis of the state of the art concluded that no present system fulfils all of these requirements to a satisfactory degree. Hence, the fulfilment of those requirements by SWeMPs would suffice as proof of value of the proposed approach.

Hence, as a means for evaluation, we return to the initial requirements analysis (first formulated in section 1.4). These requirements need to be demonstrated as met through specific tasks in the following scenarios:

1. Integrating data from at least two different sources to present the resulting composite media to the user in a meaningful way;
2. Mediating between known representations of data and knowledge and the actual representations found when retrieving from the Internet;
3. Adapting the selection of content and knowledge, as well as the presentation of media, based on the context of usage, which may be the type of user, the target device or the current usage location;
4. Retrieving additional knowledge as required to fill knowledge gaps in the process, by inferring which possible sources would be relevant to a particular aspect of the multimedia generation process;

5. Presenting multiple media objects in a composite and synchronized presentation in which the composition is based not only on low level features of the media but also on the high level representation of specific concepts to the user.

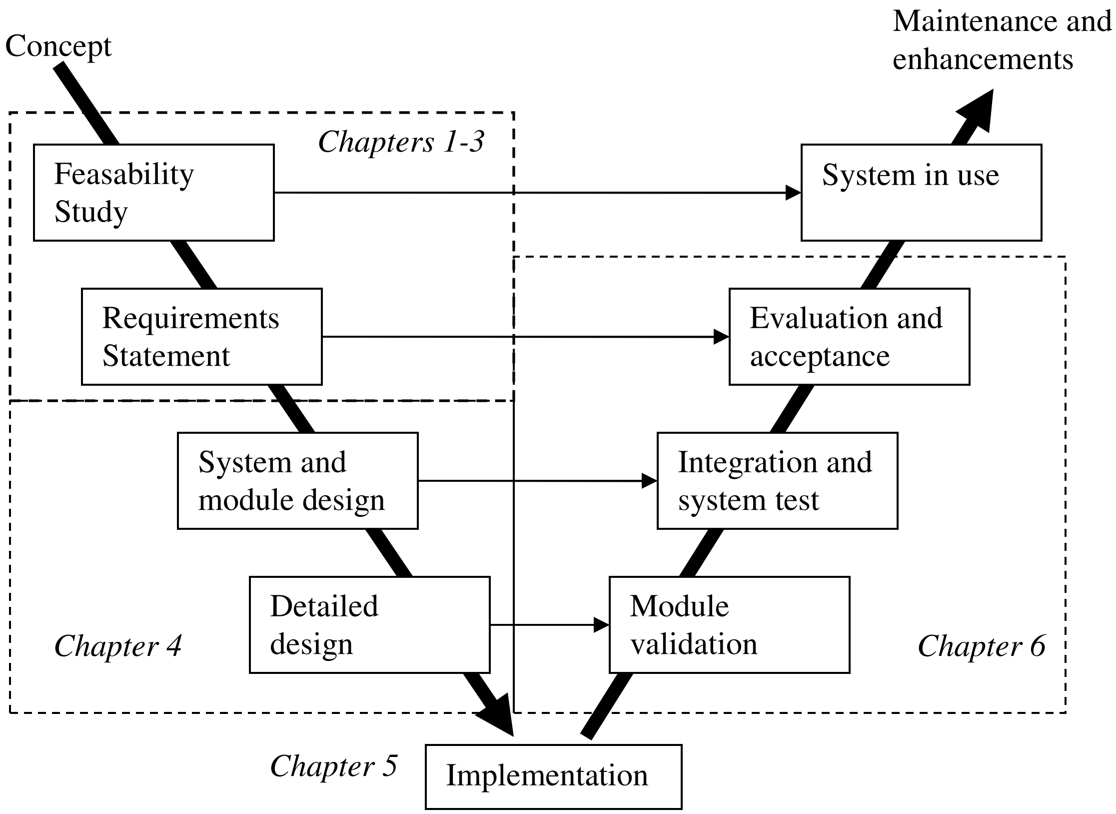

Figure 6.1 Qualitative evaluation in the software lifecycle [Tansley,1993]

6.2 The importance of domain independence

The previous two chapters have described the SWeMPs framework, both conceptually and in terms of a concrete implementation. The framework can be seen as a combination of knowledge – stored in the knowledge base and the metadata and ontologies that it references – and rules which operate upon the knowledge. In order to show how this actually enables an information request to be answered by a multimedia presentation meeting the information need, we introduce two examples of what could be called Intelligent Information Services (IIS). We note how an IIS will be realised from the SWeMPs framework through specifying the requisite knowledge, and note how the process uses this knowledge to interpret an information query and meet the information need. As a result we also want to demonstrate the domain independence of the SWeMPs approach, not only by reference to these two examples, but by extension through determining how any domain may be supported in an IIS. It is important to show that SWeMPs solves the qualitative requirements given above not only for the two scenarios that we have selected, but potentially for any scenario for which the generation of a multimedia presentation can be an effective means of interacting with the user.

The SWeMPs framework has been defined on purpose to apply to the generic knowledge-based multimedia generation task, specifying only how the task is done independently of any subject domain. To enable tasks related to a certain domain, *knowledge* about that domain, *resources* representing that domain, and *presentation rules* for communicating that domain must be added to the SWeMPs framework (and also the *services* defined which will be able to supply the additional functionality required by the application to work with the specified content and knowledge). This can be done independently of the application logic (i.e. no programming is required to create an IIS) as all of these specifics are explicitly and declaratively defined through instantiating the conceptual model of SWeMPs with references to existing data.

To reiterate, for an IIS the developer needs to specify in the conceptual model the following:

- The **knowledge** about the domain that can be leveraged by the application. We give references to ontologies which describe the domain of the IIS. We also reference knowledge sources based on this ontology relating to the concepts that the user may wish to query about.
- The **resources** that can be used to communicate information from the domain to the user. To make sure that the resource is usable, not only references to the resource itself must be made but also the resource annotation which links the resource to the concepts it can be used to represent as well as expresses the (preferred) display characteristics of the resource.
- The **services** available to the application to resolve issues such as ontological mismatches or resource adaptation. These are referenced together with their properties (service descriptions) that enable the application to select the best service available and to invoke it correctly.
- The **presentation rules** which guide the application in representing requested information to the user. These map relationships between concepts or properties of concepts, which are represented by certain resources, to communicative abstractions which are applied between those resources.

Figure 6.2 illustrates how this domain-specific content (IIS-specific) relates to the domain-independent SWeMPs framework and model which was specified and implemented in chapters 4 and 5.

The two concrete scenarios given in this chapter will illustrate how this is able to flexibly create dynamic multimedia generation processes for the domains of genealogy and tourism, and by extension, we can evaluate how the SWeMPs approach, with its close integration with the Semantic Web, represents added value for the field of Intelligent Multimedia Presentation Systems (IMMPS) for any domain.

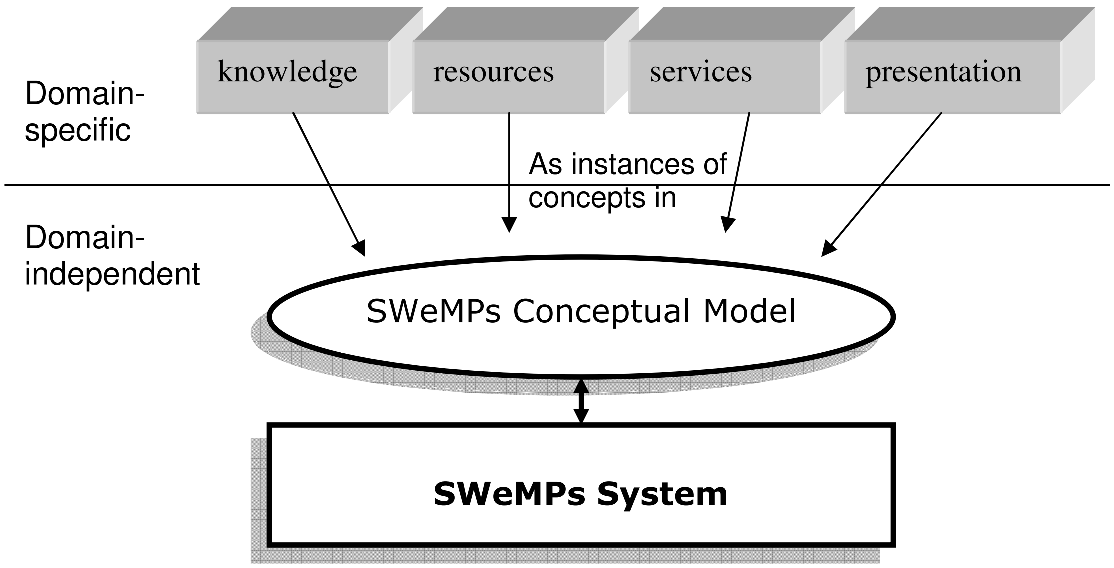

Figure 6.2 Division between domain-specific and domain-independent content

6.3 Scenario 1: A photo-based family tree

In this scenario, a user can specify a person and the service will respond with a graphical representation of that person's family tree. This representation will rely on annotated photographs of family members.

We consider which data to reference to realise this sample case:

- We locate a family tree ontology[65], based on a subset of the genealogy standard GEDCOM data model. It is in DAML+OIL, but can be trivially converted to OWL. This ontology is *referenced* from the conceptual model; when an information query refers to a term from the ontology it will *then* be retrieved and integrated into the knowledge base. The ontology supplies the following classes and properties of interest to this use case:
  Classes: `Family, Person` (with subclasses `Male` and `Female`)
  Properties: `childIn, name, surname` and `spouseIn` (with subproperties `husband` and `wife`)

- We can find or create our own knowledge using this ontology to describe a certain family. Note that knowledge could also be used which is expressed by a different ontology, together with a service which enables a mapping

[65] http://www.daml.org/ontologies/214

between the two. In this case we will express some knowledge describing a subset of a hypothetical family. This knowledge will be stored on the Web as RDF statements and referenced as metadata from the conceptual model. Some sample statements are given below, using N3 notation and the prefixes "gen" for the family tree ontology and "me" for our namespace. All subjects are instances of Individual (either the subclass Male or Female, for the properties husband and wife respectively), and all objects are instances of Family.

```
me:lars        gen:husband    me:familyNelson
me:nina        gen:wife       me:familyNelson
me:mike        gen:husband    me:familyJones
me:jennifer    gen:wife       me:familyJones
me:nina        gen:childIn    me:familyJones
me:lucy        gen:childIn    me:familyJones
me:renee       gen:childIn    me:familyJones
me:lucy        gen:wife       me:familyHughes
me:john        gen:husband    me:familyHughes
me:renee       gen:wife       me:familyRoss
me:annabel     gen:childIn    me:familyRoss
```

- We can find or create our own content for use with this service. Given the domain, we hypothesize a family photo album application which has stored images of family members together with some metadata about that image. A potential metadata format for the photos is taken from the W3PhotoSpec project[66]:
  `foaf:Image` represents the individual photo which an:annotates a RDF resource which is the URL of that photo;
  `foaf:depicts` property which can take as value a foaf:Person which must have an identifying property which can be one of foaf:mbox_sha1sum, foaf:homepage, foaf:weblog or foaf:name;
  `dc:format` property which takes as value the MIME type of the image resource;
  `svg:height` and `svg:width` properties which take as value the height and width of the image resource in pixels.

- Note that as the identification of individuals here is different from that in the genealogy ontology we will need to determine direct equivalence relations in the knowledge base between `foaf:Person`s and `gen:Person`s, so that the reasoning component can resolve the ontology mismatch. we also need a mapping from the W3PhotoSpec vocabulary to the SWeMPs conceptual model vocabulary in order to be able to relate resources to subjects.

[66] http://esw.w3.org/topic/W3PhotoVocabs

- Finally, some presentation rules need to be expressed for the family tree in terms of the genealogy ontology. These rules are made available to the rulebase in order to determine the communicative abstractions between resources, which are passed to the multimedia modeller and used to determine the family tree presentation. The communicative abstractions represent how two concepts will be related to each other in the multimedia presentation at a high level, which is then converted by the multimedia modeller into a set of applicable constraints between the resources that represent those concepts. For example, the below abbreviated[67] Prova code defines how resources representing spouses are displayed together:

```
constraint(R1,directly-left-of-AND-h-align,R2) :-
represents(R1, S1),
represents(R2, S2),
query(S1,gen:Person,gen:husband,F,gen:Family),
query(S2,gen:Person,gen:wife,F,gen:Family).
```

  Here we state that the resource representing the husband is directly left of and horizontally aligned with the resource representing the wife. Likewise we can write a rule that the resource(s) representing the parents are above and connected (e.g. with a line) to the resource(s) representing the children, and that siblings are displayed next to each other, horizontally aligned. With these simple rules we state to the application how the resources found for the presentation can be organised spatially to communicate the semantic relations between them.

Having created (where necessary) the data and referencing it in the conceptual model for the application (by declaring instances of the appropriate classes), we have a working and ready IIS for presenting family trees to interested users. We can describe the process introduced in the previous chapters (section 4.3.4, section 5.6) using this sample, referencing the data and control flow that leads the process from the initial query to a final presentation. To create the knowledge base, one could use the Protégé tool to load the SWeMPs ontology and to populate it with the instances of `Ontology, Metadata` and `Service` (with the respective `XMLNamespace`s and `Occurrence`s). As we will see, this is sufficient for the initialization of the IIS.

[67] `directly-left-of-AND-h-align` would be defined in the multimedia modeller, and could be alternatively expressed as the union of constraints for directly-next-to, left-of and h-align. The `represents` facts are directly asserted in the rulebase based on `swemps:represents` properties that can be inferred between `Resource`s and `Subject`s in the knowledge base (section 5.6 Rule 5). `query` is a rule which executes semantic queries over the conceptual model (section 5.7.1). The ontological concepts are given as Prolog terms typed by java.net.URI with the values of the respective classes and properties from the GEDCOM ontology. As we note shortly, rather than code in Prolog the presentation rule could be generated from an external SWRL file, making communicative abstractions available on the Semantic Web.

The knowledge base, in terms of instances of concepts in the SWeMPs conceptual model (i.e. the ABox of the ontology), is illustrated in Figure 6.3 (based on a visualisation by the Protégé OntoViz plugin).

As we will show in the process description, the system is able to infer from this initial knowledge the relevant subjects and resources and their respective properties and relationships. We describe the generation process in five steps, following the list of activities given in section 4.3.4.

- **Activity 1**: the user makes a request to the service for Jennifer Jones' family tree. This request is passed through the query handler component and the application receives the query in the form of an abstract statement:

```
k_sub     = *
k_prop    = {gen:spouseIn | gen:childIn}
k_obj     = me:familyJones [uri: gen:Family]
```

Here, the subject of the query is unbound and untyped, the property is a set containing the relations of being a spouse or a child in a family and the object is the concept representing the family of Jennifer Jones, typed as an URI with a value representing the concept of a Family from the genealogy ontology. In the application the subject is mapped to a free variable, the property to a Prolog list with both relations as members, and the object to a java.net.URI instance (the object type is equally mapped to a java.net.URI instance).

As well as the query, the conceptual model and the presentation constraints (e.g. user and device characteristics) may be introduced as this stage, where they take precedence over any previous model or constraints. Constraints can use any vocabulary but need to be mappable to the SWeMPs conceptual model vocabulary (for this, we extend the ontology of the ZyX model with some additional properties which apply to the root element, the ZyxModel itself).

- **Activity 2**: the application is executed by the insertion of the query from the query handler component. The first set of rules that are evaluated check the conceptual model for the namespaces used in the query and retrieve the ontologies and metadata that are associated to those namespaces. Based on this conceptual model extended with domain specific information, the query is answered, resulting in a set of possible results.

The XML namespaces that exist in the query belong to the genealogy and the user's personal namespace (in the QNames shown as the prefixes "gen" and "me"). These are the XMLNamespace instances x_1 and x_2 in the conceptual model. The ontologies which define the concepts in these namespaces (the genealogy ontology o_1) and the metadata which provides statements about

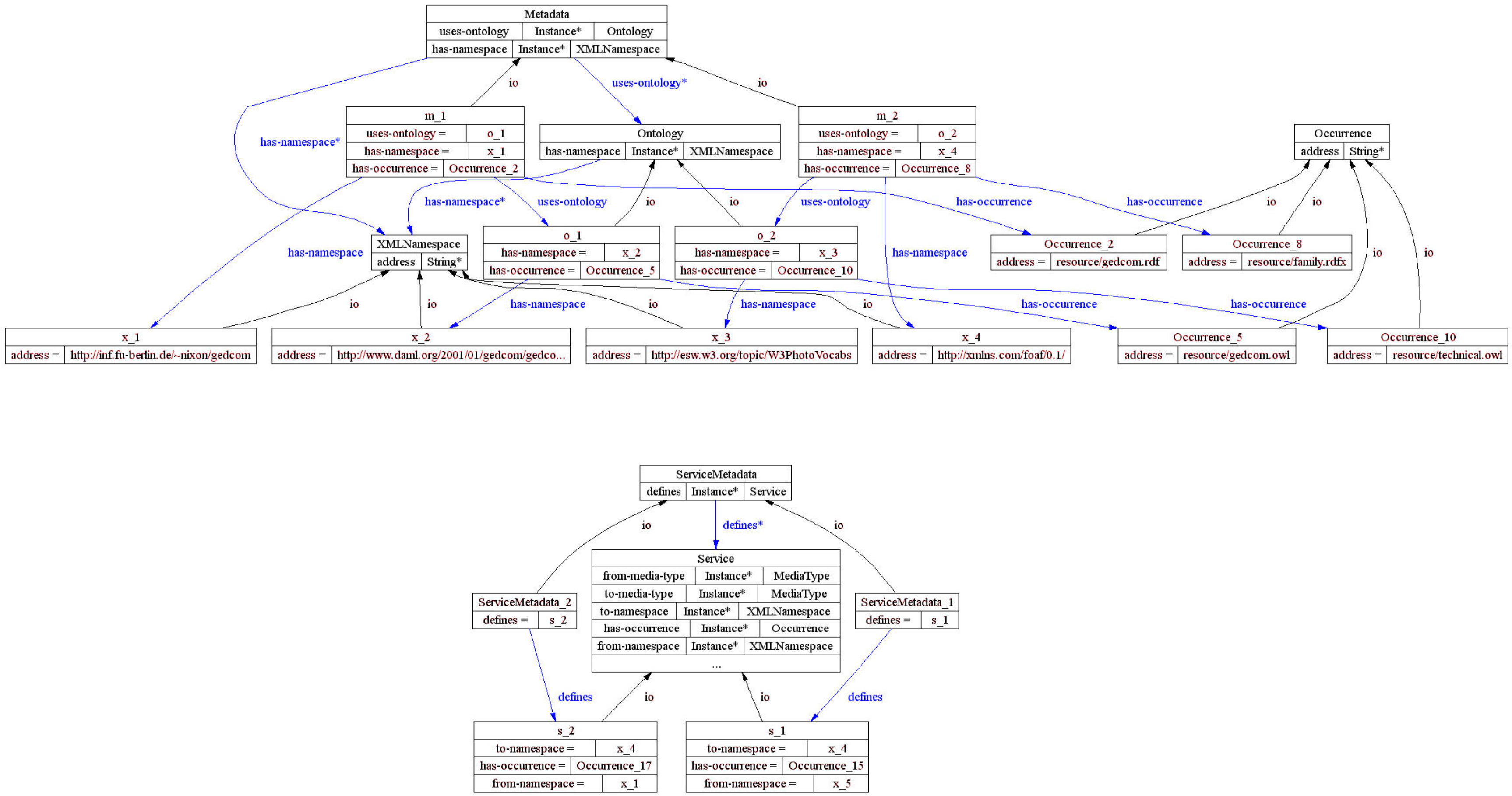

Figure 6.3 Knowledge base for Family Tree IIS

concepts in these namespaces (the author's metadata m_1) are retrieved (on the basis of has-namespace properties) and used to extend the conceptual model. Against the extended model, the query is evaluated. The dedicated query rule is able to generate a RDQL query for the Jena reasoner; the list in the property parameter means the join of two queries is to be evaluated. As RDQL doesn't support joins in the query syntax, two RDQL queries are executed and their results merged (as part of the query rule). The object type serves only to clarify that it is an URI rather than a literal (String); if the subject (the free variable) were typed, this would add a `rdf:type` condition onto the query.

```
          SELECT ?k_sub, ?k_prop, ?k_obj
(1)WHERE (?k_sub gen:spouseIn me:familyJones)
 (2)WHERE (?k_sub gen:childIn me:familyJones)
```

From the metadata m_1 the following statements will match and be returned by the reasoning component:

```
(1) me:mike          gen:husband me:familyJones
(1) me:jennifer      gen:wife    me:familyJones
(2) me:nina          gen:childIn me:familyJones
(2) me:lucy          gen:childIn me:familyJones
(2) me:renee         gen:childIn me:familyJones
```

- **Activity 3**: the application seeks resources to represent the concepts that exist in the knowledge result. This involves examining resource metadata to see which resources may represent those concepts. A mapping is generally necessary from the resource metadata format to the SWeMPs conceptual model, so that the resource properties necessary for the generation process can be extracted. For example, the relationship between a concept to be represented in a presentation and the resource which will act as that representation is expressed by the property swemps:represents. On this basis, resources for the selected concepts can be found by querying on the conceptual model:

```
SELECT ?Resource
WHERE ?Resource swemps:represents [me:mike, me:jennifer …]
```

This initial query will return null, as there are initially no statements using `swemps:represents` in the conceptual model. Rather, Rule 4b from the generation process (section 5.6) is triggered which attempts to look up metadata, and to map concepts in that metadata to the concepts of the query. A service is selected which performs look-up for `swemps:Resource`. While this could be used to extract resources from some underlying media database using a specialised approach (and hence abstracted into an external service), in this case resource metadata is available but must be mapped into the SWeMPs ontology. Hence, the service call fails (e.g. the service is not found or returns null) and the system backtracks to the other solution, which is selecting a service for mapping from the 'unknown' concept to a known (to the resource metadata) concept. Hence, services are examined that can map between the SWeMPs conceptual model and (any of) the resource metadata referenced by that model.

In this case, this is a mapping (service s_1) between SWeMPs and the ontology o_2 used by the resource metadata, which is the W3PhotoSpec photo ontology (namespace x_4). The service call uses the following mappings:

```
swemps:Resource ↔ foaf:Image
swemps:represents ↔ foaf:depicts
swemps:Occurrence/swemps:address ↔ an:annotates/rdf:resource
```

The query is now repeated using the mapping(s) that were determined, i.e. statements matching that given below will now also be found by the query:

```
?foaf:Image foaf:depicts [me:mike, me:jennifer …]
```

The query result is still null, as the metadata specifies the depiction of `foaf:Person`s (hence the FOAF namespace x_3 is linked with the resource metadata in the knowledge base) and the object of the query uses `gen:Person`s identified in the author's namespace. Again, an attempt is made to first extract new metadata about the concepts `me:mike, me:jennifer` … and then to determine mappings from that namespace to individuals in other namespaces which are used by metadata in the conceptual model.

For example, a service for knowledge extraction is sought for the concept `me:mike`. We could use a Web service interface to a Semantic Web-based FOAF repository in which FOAF properties are added to concepts representing people to allow their identification Web-wide (service s_2). Based on the unique property characteristic of `foaf:mbox`[68] the Web service can infer from this repository and hence return as a result:

```
me:mike owl:sameIndividualAs [your:mike2, other:mike, …]
```

These mappings permit the query to be resolved by metadata using other vocabularies. From the resource metadata (the photo album annotations file) we find and assert in the system working memory a set of resources which represent the concepts in the results set:

resource(album:img12), resource(album:img24) …
represents(album:img12, me:mike), represents(album:img24, me:jennifer) …

Additionally, the resource properties are checked against the presentation constraints, using again knowledge mapping to resolve the use of different ontologies. Here, the system unites the ontologies used in the resource properties (the W3PhotoSpec vocabulary) and the presentation constraints (in this implementation, an extension of the CC/PP vocabulary) in the ZyX model used by the multimedia modeller. A mismatch has been illustrated in section 5.6 rule 6, and the use of a service to convert the resource to meet the presentation constraints has been described in section 5.4.

[68] This is based on the OWL InverseFunctionalProperty. As this can only be applied to datatype properties in OWL Full, it is not possible to extract `foaf:mbox` statements and then use the SWeMPs reasoner to infer equivalence (as we assume OWL DL for computability). Hence this step is handled by the dedicated service. Open source code is available for this task, see http://www.hackdiary.com/archives/000021.html

- **Activity 4**: The selected resources are passed to the multimedia modeller, together with their properties. The modeller determines additional constraints between the resources from the communicative abstractions that can be inferred from the semantic relations between the concepts represented by those resources.

Five resources (one for each of the concepts returned by the query) are added to the multimedia model. Three communicative abstraction rules have also been provided, and we consider how resources are constrained in their presentation on the basis of relationships between the concepts they represent.

These rules could be encoded into an exchangeable syntax such as SWRL and placed on the Web, and hence retrieved by the system in the same manner as ontologies; currently for the prototype we map SWRL into the native Prova syntax through a dedicated transformation and add the generated rules to the rulebase so that they are automatically evaluated as representation facts (i.e. the represents predicate) are added to the working memory. The constraints are extracted by unification on the inserted constraint rule, i.e. a constraint of type T applies to resources R1 and R2 when the rule constraint(R1,T,R2) evaluates in the rulebase to true.

The constraint rules used in this scenario, with some OWL individuals from this scenario unified to the variables, are given below:

represents(album:img12, me:mike), represents(album:img24, me:jennifer), query(me:mike, gen:husband, me:familyJones), query(me:jennifer, gen:wife, me:familyJones)
→ constraint(album:img12, ft:directly-left-of-AND-h-align, album:img24)

represents(album:img33, me:nina), represents(album:img24, me:jennifer), query(me:nina, gen:child, me:familyJones), query(me:jennifer, gen:wife, me:familyJones)
→ constraint(album:img33, ft:below-AND-connected, album:img24)

represents(album:img33, me:nina), represents(album:img52, me:lucy), query(me:nina, gen:child, me:familyJones), query(me:lucy, gen:child, me:familyJones)
→ constraint(album:img33, ft:directly-next-to-AND-h-align, album:img52)

To explain these inferences to the communicative abstractions, the first inference states that Mike and Jennifer, as husband and wife, are displayed with Mike directly to the left of Jennifer at the same horizontal level (which means taking the same y-axis value). Secondly, that their children Nina, Lucy and Renee are displayed below the image of Jennifer and are connected by a line. Thirdly, that the three siblings are displayed directly next to each other at the same horizontal level.

In the implementation, the constraints are added to the multimedia model using the insert rule (section 5.7.3). These are mapped from URIs (as unique identifiers for a communicative abstraction) to executable objects (Java methods applicable to a ZyX MediaItem) which define the constraints that they represent. The mapping uses a property file which is part of the multimedia modeller, and the executable objects are coded according to its underlying implementation (e.g. the use of the ZyX model and Cassowary constraints within Java).

- **Activity 5**: From the complete set of constraints, a finalised model is determined (i.e. a model in which all resource properties are bound). This model is formatted and delivered for presentation to the user.

This maps to a spatial layout like the one shown below (Figure 6.4). All constraints are solved, note that where no constraint exists, presentation has to be determined by the modeller e.g. in this case the left-to-right order of the three siblings.

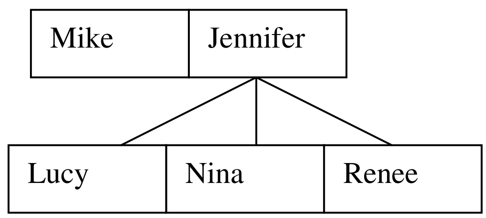

Figure 6.4 Final layout of a Family Tree result

In this scenario, we considered the possibility of requesting the family tree from a wireless mobile device. A suitable final format for display would be SVG. Using a XSLT transformation from the XML serialisation of the abstract model produced in the system to SVG (encapsulated in SWeMPs by the formatter component), a resulting presentation could take the form below (Figure 6.5)[70]. This SVG is compatible with the SVG Tiny and SVG Basic specifications [W3C,2003], meaning it could be displayed on a mobile device such as a cellular phone or PDA.

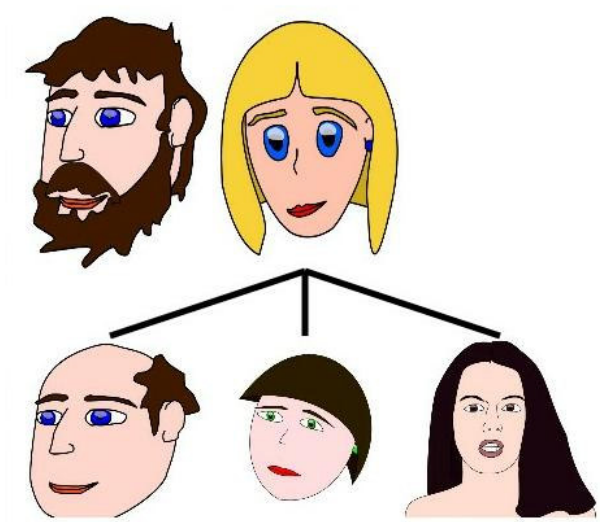

Figure 6.5 SVG result of the Family Tree scenario

[70] Images used were free clipart from http://clipart.freedesktop.org

6.3.1 Summary

The above scenario description has illustrated the multimedia generation process, as proposed in the SWeMPs framework. It is significant in that it uses both implicit and explicit knowledge and semantics, together with reasoning and rules, to answer information queries intelligently and deliver information results coherently using multimedia presentation. Only a small set of data was used to be able to clearly illustrate the workings of this scenario, but it should be clear that this can scale (the only limit being the performance of the rule base and the other components).

We can assess the effort of implementing this scenario in terms of the external (domain-specific) content which must be provided to the generic (domain-independent) SWeMPs framework and model.

		ONTOLOGY (2)	METADATA (2)
Knowledge		Re-use genealogy	Created by hand (11 statements in 1 file)
Resources	Re-use existing digital photos	Re-use W3PhotoSpec	Created by hand (8 foaf:Images in 1 file)
Services	2 Service instances: Instance mapping (foaf:Person <-> gen:Person); Vocabulary mapping (SWeMPs <-> W3PhotoSpec)	None required	Dedicated SWeMPs properties on Services are used (section 5.4); Optional ServiceMetadata can provide additional QoS
Presentation	Created by hand (3 SWRL rules)	SWRL could be used	None required

It can be seen that for the family tree scenario, the effort required is quite small. Re-use of existing ontologies and resources is shown. The metadata is created by hand; this is a result of the lack of real world Semantic Web usage at present that means the average user does not have RDF at hand which they can use in SWeMPs like applications. In the hypothesized future, where home networks annotate and organize family data, we could expect that genealogical data about the family is available to the system (parents may like to see what their children are doing; siblings might want to share data with one another) and photos of family members are annotated. Services and presentation rules are the only aspects which would potentially still require preparation in the development of this Intelligent Information Service, and even then once such services and presentation rules are available they could be published with their descriptions on the Semantic Web and found by other (SWeMPs) applications.

While this scenario could confirm that the proof of concept system functions as has been specified in Chapter 5, it is too simplistic for a proper evaluation of the effectiveness of the approach. Hence in the following section we turn to the second scenario that has been outlined in this thesis.

6.4 Scenario 2: Interactive tourism television

In order to demonstrate a real world viability of the SWeMPs approach by handling the issues of Semantic Web integration with multimedia presentation, we now turn to the more detailed scenario based around a tourism information service (refer to section 1.3.2). We introduce the data that will be prepared for or re-used by the scenario, and illustrate (using the three sub-scenarios that were given in section 1.3.2) how the SWeMPs framework meets the requirements of IMMPS that have been detailed in section 6.1.

For this scenario, we omit the detailed description of the process (which should be clear from this previous scenario) and focus on how SWeMPs meets the requirements of a next generation of Intelligent Multimedia Presentation Systems.

For the evaluation, we consider how each sub-scenario demonstrates meeting particular requirements of the system.

To reiterate, this scenario is intended to demonstrate the real world operation of the SWeMPs approach, and the potential advantage of multimedia presentation for information communication where that information and its presentation is inferred from the global distributed knowledge store of the Semantic Web. In that light we focus here also on re-using the existing data and knowledge sources on the Web.

6.4.1 Initial preparation for this scenario

As has been shown in section 6.2, developing an intelligent information service (IIS) with the SWeMPs framework requires the definition in a knowledge base using the SWeMPs conceptual model (together with respective locations and characteristics) of:

- The **knowledge** available for use in that IIS
- The **resources** available for communication of a response from that IIS
- The (Web) **services** available for performing specific tasks within that IIS
- The presentation **rules** available for determining a final presentation for that IIS

For the touristic scenario, we decide on a concrete domain to focus the implementation on and determine the sources which are available for building that service. In our initial considerations, knowing of the presence of food and wine ontologies, we decided to build an IIS upon the theme of restaurant cuisine. A visitor to a city is likely to want to eat out (a hotel guest has little other choice). We consider this a good exemplary scenario as unlike other specialised domains, we all need to eat, have preferences that we wish to have taken in consideration and would welcome a guide to the best restaurants (according to our preferences) in the cities we travel to.

To create the Semantic Web knowledge about restaurants for a given city, we face the typical “early adopter” problem: there is no freely available RDF describing restaurants in a city. However, there are Semantic Web tools which offer the opportunity to “screen scrape” knowledge out of syntactic data such as HTML web pages. Piggybank[71] is a means to generate RDF from HTML web pages, albeit through manually authored screen-scraping code. This is an imperfect solution as it is generally only applicable to the single web page it is written for, and will break as soon as any (significant) aspect of the HTML structure changes. On the other hand, large content providers tend to employ a common structure across pages which deal with the same topic (e.g. restaurant listings in a travel guide) and may be less likely to change their structure on a whim (although site redesigns can not be ruled out!).

Two further issues arise concerning restaurant knowledge. One is a suitable ontology to express that knowledge, and the other is the association of relevant resources to the knowledge that is being generated. We found a gastronomy ontology[72], and will use it for the purposes of the scenario. Secondly, we aim to not only add the resources that can be scraped from the Web using PiggyBank, if any, but also to integrate relevant resources from other sources. This has the added value of metadata availability, which will not be the case with images scraped using PiggyBank (as HTML tends to simply contain an IMG tag, without further information). We will have to consider a source of images which carry some form of metadata and choose Flickr[73] which is a web site that contains tagged photos submitted by its members.

Considering what sort of information we can relate to the restaurants in order to build a useful IIS, we decide on representing the address and the establishment's cuisine. The latter is expressed in the chosen gastronomy ontology, as a set of subclasses of a Cuisine concept, each of which has an instantiated individual[74]. For the address, we extend the ontology with the properties address and city which take String literals as values. There is also a model for geophysical location[75] which is able to express latitude and longitude in a way that can be passed to various useful services on the Web. Maps for communicating location could be generated by a source such as Yahoo Maps, which offers a RESTful Web Service interface[76].

There remains the issue of the tourist video material itself, and we choose to use a MPEG-4 video available from the Open Video Project[77] which promotes tourism in Coral Gables, Florida. The three minute program "shows colourful parrots, the University of Miami, and a visit to the beach". It dates from 1950, but it is satisfactory for demonstration purposes and is copyright free. We use the IBM MPEG-7 Annotation tool[78] to create a semantic annotation for the video, drawing on instances from a sightseeing ontology. We use the OntoBroker tourism ontology[79] which defines a concept Sehenswuerdigkeit (the German word for a tourist sight).

[71] http://simile.mit.edu/piggy-bank/index.html
[72] www.csd.abdn.ac.uk/research/AgentCities/ontologies/restaurant-v4.daml
[73] A leading example of what has been termed "Web 2.0" due to its social networking and tagging aspects. http://www.flickr.com
[74] This is a usual modeling "trick". As the object of a statement can not be a class in OWL-Lite/DL, a single instance of each class is created which can be used as an object in a statement in place of the class itself. See the discussion on "Representing Classes As Property Values on the Semantic Web", W3C Working Group Note, 5 April 2005 http://www.w3.org/TR/swbp-classes-as-values/
[75] Basic Geo (WGS84 lat/long) Vocabulary http://www.w3.org/2003/01/geo/
[76] http://developer.yahoo.net/maps/
[77] http://www.open-video.org/details.php?videoid=4250
[78] http://www.alphaworks.ibm.com/tech/videoannex
[79] http://ontobroker.semanticweb.org/ontos/compontos/tourism_I3.daml#

What remains is to check for services that will map between the different ontologies used according to which data will need to be integrated in the multimedia generation process and that will adapt resources as required (e.g. transcoding or scaling images). Finally, presentation rules are defined in terms of representative ontologies to determine the presentation of the resources in the final result.

6.4.2 Extracting the data

To acquire the semantic data from the current Web we use a mix of approaches which indicate the viability of SWeMPs despite the lack (today) of a widespread Semantic Web.

6.4.2.1 Yahoo Travel – screen-scraped

In the current Web, scraping knowledge from Web sites is still a relatively time consuming and manual process. While automated screen scraping is indicative of how the "Syntactic" Web could migrate to the Semantic Web, this is based at present on two observations:

- In the PiggyBank approach, a user community writes JavaScript or XSLT for specific websites. JavaScript can parse the HTML web page DOM even though it is not (XML) well formed, while in this case XSLT will break. Both can be rather non-trivial to write: JavaScript requires programming skills and XSLT is not a fully fledged programming language which means some actions require complicated approaches. A key issue with this approach is that content is extracted without the explicit permission of the content owner, which – copyright issues aside[80] – means control over the content remains with the owner, who can alter the HTML code at will and break the scrapers that have been written.
- A more ideal approach, which can only be hoped for, is that the content owners – in the same way that some are making their content available over Web Service APIs for use in other Web sites or Web-based applications – will choose to empower the Semantic Web by making their content available in structured formats (X(HT)ML at least) and using a standardized approach like GRDDL[81] to link content to their own scraping code (e.g. XSLT) which produces RDF for consumption by Semantic Web systems such as SWeMPs. In this case, the content owner maintains control over the knowledge that they make available as well as guarantee its availability.

Given the lack of content at present in the second approach, we have followed the first approach. All of the RDF was generated from Yahoo Travel web content. As different objects each carry a numeric ID in their URI, there is an existing identification scheme that can be reused. For the key sights of a location, there is a RSS feed from which RSS items can be mapped into `t:Sehenswuerdigkeit` ("t" is the QName prefix for the tourism ontology) instances with titles mapped to rdfs:labels and links mapped to RDF resource URIs (we take a substring as we require essentially the Yahoo Travel domain and the unique numeric ID). The remaining content had to be extracted from the HTML, which requires two preparation steps

[80] Sites such as Yahoo!, Amazon and eBay are discovering the value of making their content available through APIs and mixed with other content on third party sites, and tolerate this within certain restrictions (such as non-commercial use) because it leads to the further dissemination of their content on the Web.

[81] http://www.w3.org/2004/01/rdxh/spec

(tidying up the HTML and then converting it to XHTML[82]), followed by a transformation using a dedicated XSLT stylesheet. From this, we acquired `r:Restaurant` instances as well, extracted address strings for all of the concepts, and added a `r:typeOfCuisine` relationship between `r:Restaurant` and `r:Cuisine` instances. This results in a core domain knowledge base for the tourism application.

6.4.2.2 Video annotation

The IBM MPEG-7 Annotator automatically makes a segmentation of the provided video along with key frames for each segment. It uses a XML-based lexicon (a MPEG-7 ClassificationScheme) which provides MPEG-7 terms for annotation. These terms have a local ID and a plain text Name. The tool generates a MPEG-7 XML file which contains the temporal decomposition of the video into VideoSegments. Each segment has a MediaTimePoint and MediaIncrDuration which identify the time point in the video in which the segment begins and the duration of the segment. Annotated segments have a TextAnnotation element which contains the name of the MPEG-7 term as a FreeTextAnnotation. To adapt the MPEG-7 XML for use with semantics, we had to carry out the following steps:

- The VisualSegments in the MPEG-7 file require unique IDs. Local IDs were generated based on the start time of each segment (which is necessarily unique within the document).
- The ClassificationScheme also requires allocating Semantic Web URIs to its terms. To do this within the MPEG-7 standard, we use ControlledTermUse elements for the annotation instances that have been defined, and their Semantic Web URIs (drawn from the populated ontology in which they exist) for the href attribute value.
- Controlled terms are used by the MPEG-7 Semantics DS in the Label element of SemanticBase. So the terms with which the video is annotated are 'instantiated' in the MPEG-7 document as ObjectTypes (which is an extension of SemanticBase) whose Label href is equal to the ControlledTerm URI. From this step, it is clear that the ClassificationScheme could be omitted if the annotation tool permitted the direct inclusion of Semantic Web ontologies.
- Finally, the MPEG-7 document is also extended with Relations of type SegmentSemanticBaseRelationType (i.e. they relate between a media segment and a semantic object) with the name attribute taking the value hasMediaReferenceOf, which defines the property of depiction in MPEG-7. In each case, the source is an ObjectType and the target is a VideoSegment.

Figure 6.6 shows a screenshot of the annotator. A lexicon with Yahoo Travel categories has been added by manually creating a XML file based on the categories of the Yahoo Travel website. If those categories were available in a structured taxonomic form it would be possible to produce the lexicon through a transformation, while ideally the annotation tool would accept RDF/OWL as input. Attaching concepts to a segment of video is as simple as ticking the box for a chosen segment. The

[82] We used the online converter http://www.it.uc3m.es/jaf/conversor.html

output is a standard MPEG-7 file which links lexicon concepts to video segments but without the associations to Semantic Web URIs that were added in the lexicon.

Clearly, it would be desirable for a MPEG-7 annotation tool to produce a Semantic Web-friendly result. As the IBM MPEG-7 annotator is not open source, it is not possible to edit its code so currently we have to manually edit the XML lexicon and resulting MPEG-7 annotation. It would also be possible, given the connection of MPEG-7 terms to Semantic Web URIs in the lexicon file (which the tool does allow to edit manually), to write a transformation that could then map a MPEG-7 file produced by the tool to the semantics-enabled version described here.

Figure 6.6 IBM MPEG-7 Annotator being used to produce MPEG-7 with Semantic Web concepts based on Yahoo Travel categorization

A further evaluation area would be using the MPEG-7 ontologies[83] to express the annotation in RDF/OWL. For example, Jane Hunter proposes an integration of ABC and MPEG-7 ontologies such that MPEG-7 MultimediaContent depicts ABC Items [Lagoze,2001]. However, as we want to show support for the existing standards as much as possible in this “real world” scenario, it seems more reasonable at present to generate the standard-conforming MPEG-7 XML file and map directly to the SWeMPs vocabulary internally in the multimedia generation process through a dedicated Web service.

[83] See http://archive.dstc.edu.au/RDU/staff/jane-hunter/events/paper.html and http://www.acemedia.org/aceMedia/files/resource/eswc05.pdf

6.4.2.3 Getting resources and resource metadata

There are some interesting initiatives (regardless, at an early stage at the time of writing) to generate useful semantic annotations for Web based content such as text and images. We will consider two approaches:

- Adding semantic markup to the open source Wikipedia system so that Wiki entries can be annotated;
- Generating RDF metadata for photos that are published online by Flickr.

While there is some discussion at present on introducing semantic markup to Wikipedia so that articles could also contain properties and relationships that are exportable as RDF/OWL[84], the present MediaWiki system is purely syntactic and uses its own markup.

We choose to generate annotations for selection of relevant articles and their usage at different granularities of text and media. As a basis, we use the structure of the provided templates at Wikitravel[85]. We create articles in Wiki markup following the appropriate template and then scrape RDF from that markup, using hidden relations and attributes to express RDF properties. As this is non-standard, we have not edited directly the online pages at Wikitravel but made local copies with the hidden relations and attributes added, in order to demonstrate the viability of the approach without changing Wikipedia itself (including that it does not impact on the human readability of the Wiki articles, while enabling their machine processability). For example, locating Coral Gables in South Florida could look like this:

```
Coral Gables is a city in [[is located in::South Florida]].
```

In the envisaged SemanticMediaWiki, the article name after the :: (property relation) would be recognized and a link would be created. A RDF crawler could however extract from this something like:

```
<wiki:Coral_Gables, wiki:is_located_in, wiki:South_Florida>
```

A script can take this markup and output the internal annotation of the article as RDF triples. This could be re-usable with minor changes in some future SemanticMediaWiki.

As an additional approach which does use the online articles without altering them, we assume that a sufficient textual description of a concept can be extracted from a Wikipedia article by taking either the first section (where the article is split into sections) or the first paragraph (where not). The online Wiki markup can be accessed from the textarea HTML element in the 'edit' page of the Wiki article (the standard URL structure is http://en.wikipedia.org/w/ index.php?title=*{insert article name here}* &action=edit). This could be 'scraped' to produce a reasonably short text resource for the concept.

Flickr permits the open publication of photos and supports a simple categorization of its content through user supplied tags, which are freely chosen text strings. While not as rigorous as a controlled vocabulary, this simpler approach (as users do not need

[84] http://meta.wikimedia.org/wiki/Semantic_MediaWiki
[85] http://wikitravel.org/en/Wikitravel:Article_templates

to try to align their perception of the photos subject with a set of predefined concepts) has proven popular, also in many other sites that collect data (e.g. blog entries in Technorati or bookmarked sites in del.ic.io.us). A Flickr2RDF service[86] is available that generates RDF annotations from Flickr photos. Tags are given URIs by appending the string to the namespace http://www.flickr.com/photos/tags/ and related to the photo by the foaf:topic property.

Searches by tag provide a set of results both as Piggy Bank scrapable XML and as a RSS feed. The URI of these results can be automatically determined as they follow a standard format. The RSS feed adds some useful information not in the Flickr2RDF service: image format and size, as well as the URL of a thumbnail version of the image. However the RSS 2.0 used is not fully valid RDF and this feed would need pre-processing to be able to be used by SWeMPs. What is also available directly from the Flickr search results page (and can be scraped from the Web using PiggyBank) is a simpler set of image properties, including a link which points to the web page of the image and subject which contains the tags belonging to the image. This is sufficient for a pre-selection of images, whose URLs could then be fed to the Flickr2RDF service to extract their (RDF) metadata.

A key aspect for the resource metadata from Flickr is mapping the (free text) tags to (Semantic Web) URIs. The Flickr2RDF service makes simply an one-to-one mapping from tag to (Flickr-namespaced) URI, i.e. does not connect tags with any concepts from domain ontologies. We propose to handle this in that we assume combinations of tags are sufficient to uniquely identify what would otherwise possibly be ambiguous subjects, e.g. the tag "*venetianpool*" combined with the tag "*miami*" allows the system to conclude that the image depicts the Venetian Pool in Coral Gables, FL (effectively a Miami suburb) as opposed to some pool in Venice, Italy. This disambiguation is the main requirement of manual preparation that must be done in extracting RDF from Flickr. For example, we extract matching image data from a search for photos tagged with both *venetianpool* and *miami* and find 3 images[87]. Each images' metadata is extracted as RDF through Flickr2RDF. Given the rule:

```
_x foaf:topic http://en.wikipedia.org/wiki/Venetian_Pool
     ←
_x foaf:topic http://www.flickr.com/photos/tags/venetianpool
_x foaf:topic http://www.flickr.com/photos/tags/miami
```

a basis is established to map between Semantic Web concepts instantiated in terms of some subject domain (here, using the URL of a Wiki article as an unambiguous identifier) and Flickr tags in the Flickr2RDF service.

This could be automated in that a service takes a String value from the chosen concept (taken from typical properties such as rdfs:label or dc:title[89]), retrieves the Flickr search results for tags with that string and converts the metadata of the image

86 http://purl.org/net/kanzaki/flickr2rdf
87 Search made on 10 February 2006.URL used http://www.flickr.com/photos/search/tags:venetianpool,miami/tagmode:all
89 Naturally this is not perfect; for example "Crandon Park Beach" finds no photos, but "Crandon" with "park" would return one image, and "Crandon" with "beach" returns seven images (search of 29 September 2005)

set to RDF. The SWeMPs framework then repeats its query for resources that represent the chosen concept and will retrieve the Flickr photos provided:

- There is a mapping of owl:equivalentProperty between foaf:topic and swemps:represents
- There are rules mapping between a set of Flickr tags and the chosen concept (disambiguated through the presence of tags other than the tag for which the initial search was made).

There is a modelling difficulty in RDF/OWL with establishing the equivalence of an individual with the intersection of two or more individuals (it is only possible with classes). Hence the Flickr<>other mapping will require a dedicated rules-based service which reads in mapping rules as e.g. SWRL.

6.4.2.4 Resources from existing Web services

Finally, we will use dynamically generated maps from the Yahoo! Maps service, making use of the Map Image API that is available to retrieve single map images. This uses the URL http://api.local.yahoo.com/MapsService/V1/mapImage appended by a set of parameters as defined by the API[90].

The URL needs to be generated by the system. Fundamentally it contains the address of the object to be shown on the map, which could be replaced by geolocation information (longitude/latitude), as well as settings for image size, format and map zoom level.

6.4.2.5 Knowledge Acquisition through Services

It would be hoped as the Semantic Web grows, that RDF based information will become common and that content providers will realise the benefit of making their content available in RDF format. While things like RSS feeds are a beginning, or website APIs which return XML, this is insufficient for automated processing with unambiguous concepts as is required by the advanced multimedia generation capabilities of the SWeMPs framework. To bridge this semantic gap, RDF is generated from websites using services, often either JavaScript/XSLT based screen scraping or API based XML retrieval which is then mapped to RDF. This is the current approach in the Semantic Web community to solving the information problem (i.e. that Semantic Web knowledge is much more usable by applications than HTML or XML, but that this knowledge is not presently "out there" on the Web). Hence it seems appropriate to mention here a few service-oriented approaches to knowledge extraction which could produce RDF for consumption in the SWeMPs framework for multimedia generation:

- *GRDDL* is a W3C draft[91] which seeks to standardize a transformational approach to knowledge extraction from documents in XML or XHTML. In XHTML a profile is added to the head element to indicate the presence of meaning-preserving transformations, which are referenced in link elements. These transformations take the source document as a single input and output a RDF representation of content in the document. Typically XSLT is used as the transformational language.

[90] See http://developer.yahoo.com/maps/rest/V1/mapImage.html for full details of the Map Service API
[91] As of 25 August 2005

- *Microformats* is the name given to an approach to add machine readable data within human readable formats through the deliberate use of dedicated attributes in elements to unambiguously differentiate markup. This has led to approaches to generate RDF from microformat content. GRDDL has been proposed as a common approach to specifying how to handle microformats.

The important aspect in this approach as opposed to screen scraping is that here the content owner controls the process (as the changes are made in the content itself), can determine the microformat content used and specifies which transformations are available for his/her content. At present we use a very dedicated, and hence not very re-usable, manual process which encapsulates a content user-focused transformation in which HTML is selected, prepared, converted to XHTML and mapped with a specialised stylesheet.
However, a future GRDDL-conforming service would be a means to ensure that RDF could be automatically generated from Web content conforming to the content provider's wishes. For example, if Yahoo Travel pages were XHTML compliant and contained a GRDDL-compliant link to a dedicated stylesheet, the service could accept as single input the XHTML document, retrieve the XSLT and execute the transformation internally, and return the RDF result. The RDF would use a vocabulary chosen by Yahoo Travel and published publicly so that applications could be written for consuming it. The GRDDL service could be used with all conformant Web sources.

6.4.2.6 Resource Acquisition through Services

In this area, the situation has improved greatly with the approach of many Web sites to publish APIs through which applications can access their content. In this scenario we have considered two sites which make an API available: Yahoo Maps and Flickr. There does not seem to be any MediaWiki API, instead we find there is a consistency in the URL structure that can be used to retrieve the Wiki markup from an article and have written our own scripts for scraping both semantics from markup (following the proposal for SemanticMediaWiki) as well as text from the (syntactic) Wikipedia.

The API calls or Wiki retrieval & parsing code is encapsulated into external Web services which are made available to the SWeMPs framework. We should note that the Web services generally use non-semantic content as both input and output and hence the Web service call must be composed with other services which mediate between the service input and output and the correct input and output for SWeMPs:

- For Yahoo Maps, the service uses the Map Image API to fetch a reference to a GIF image by sending the request to an URI which specifies the location(s) to be displayed. The URI has to be generated from the RDF that is available to the system.
- For Flickr, a string input is passed directly to the search service and all photos with a matching tag are retrieved. For the matching photos, the photos' RDF metadata is extracted using Flickr2RDF (which itself is a Web service) and returned as a single document to SWeMPs. A further service is necessary to perform the reasoning over Flickr tags as OWL is not able to derive new facts from multiple existing facts, which requires a rules-based approach.
- For Wikipedia/WikiTravel, the URL of the Wiki article can be used as input as we use the same URL as the Semantic Web URI identifier for the concept

described by the article[92]. The WikiTravel service returns a RDF document, the Wikipedia service a text resource. To provide necessary concept mediation, a mapping is also necessary for between Yahoo Travel URIs and Wikipedia URIs. This could be included as an OWL ontology containing owl:sameIndividualAs statements.

The details of the internal coding of the Web services are not necessary to give here, though it is worth noting their re-usability in other contexts other than SWeMPs.

6.4.2.7 Knowledge Adaptation

The principal aim of knowledge adaptation is to provide mappings between different ontologies. While this can be a simple case of stating equivalences, which is possible directly in OWL at the TBox level, semantic matching can also be less trivial (particularly at the ABox level). In this scenario, there is one case where a service is necessary to map between Flickr and Wikipedia URIs which refer to the same concept. Different approaches are possible to determine in an automated fashion the equivalent between two instances, including the use of unique properties (e.g. a person's social security number), natural language analysis and clustering with related terms[93]. However, unique properties are represented in OWL by InverseFunctionalProperty which can only be applied to datatypes in OWL Full, and since we want to retain reasoning tractability we are restricted to OWL DL. Many instances also do not have unique properties to identify them and equivalence determination could require dedicated algorithms which would be wrapped as Web Services and made available to Semantic Web applications such as SWeMPs. Flickr's use of tag clusters to allow for the disambiguation of words through their association with other words (e.g. http://www.flickr.com/photos/tags/chips/clusters/) is a step in this direction and similar research in the Semantic Web community [Maedche,2002] could form the basis of "instance mapping" services.

A dedicated case of ontology mapping for SWeMPs regards the resource metadata. A common description is required internally by the SWeMPs framework to be able to reason over resource characteristics and determine constraints for the multimedia modeller. We have defined a common vocabulary drawn from the SWeMPs and ZyX ontologies. Typical resource metadata vocabularies need to be mapped to these ontologies, and this mapping is encapsulated in OWL or SWRL.

6.4.2.8 Resource adaptation

The task of resource adaptation can be expressed semantically in reference to the common SWeMPs/ZyX ontology. The service itself encapsulates local code for accepting a resource and a specification of the adaptation to be done, and returns an adapted version of that resource.

---

[92] It is widely recognised as good practise to use an URL which provides a machine retrievable human readable disambiguation of a concept as the unique identifier for a concept in the Semantic Web.

[93] Interestingly, ontology mapping research focuses on the TBox, i.e. classes and properties. Yet in a field such as multimedia presentation generation, determining instance equivalence is an important requirement and hence we can identify this as an area requiring further research.

As a demonstration of service-based content adaptation, we encapsulate a program for image transcoding (using the Java extension `javax.imageio`) as a Web Service and make it available to the SWeMPs system. Further media adaptation can be encapsulated in services written in any implementation language, e.g. the Java Media Framework which provides a wide range of methods for audio and video manipulation.

6.4.2.9 Resource presentation

Presentation rules are written in SWRL and relate domain specific properties to communicative abstractions (sets of presentation constraints). Authors can use the set of core abstractions available in SWeMPs, but are also able to write their own if they wish. Abstractions are created as Java methods which use the multimedia modeller and Cassowary Java classes to specify constraints on the multimedia model – this will be documented with the implementation. The properties file for the multimedia modeller – which maps constraints (identified by URIs) to the Java method to be called – must also be updated in the system to allow it to use the user-defined communicative abstractions. For this scenario we also have to consider the various spatial, temporal and interactive constraints that may be used. Most presentation can be governed by a combination of the core abstractions such as:

> *Spatial:* above, below, left of, right of, in top left corner of, …..
> *Temporal:* before, after, during, start with, end with, …..
> *Interactive:* appear on click, disappear on click, …..

These are related in the scenario to knowledge about the concepts of the presentation and the resources representing them. Typical predicates in the body of the presentation rules are:

- Check instance of which ZyX `MediaObject` type
- Check (non-)equality of two concepts
- Check property of representation (`swemps:represents`) between a concept and a media object
- Check resource metadata property between media object and a value
- Check conceptual metadata property between a concept and a value

6.4.3 The conceptual model

Given the existence of the chosen sources of knowledge, content, (Web) services and (presentation) rules, an implementation of the SWeMPs framework for a particular task is realised in that the SWeMPs ontology is populated with instances of `Ontology`, `Metadata` and `Service` (together with `XMLNamespace` and `Occurrence` instances, as necessary). The instances used are enumerated below:

Ontology ID	Title
o_1	Gastronomy
o_2	Tourism
o_3	MPEG-7
o_4	Yahoo Travel (only ABox)
o_5	Wikipedia Travel
o_6	Flickr

Metadata ID	Title	Ontology	Namespace
m_1	Coral Gables Restaurants	o_1	o_4
m_2	Coral Gables Sights	o_2	o_4
m_3	Tourism Video Annotation	o_3	o_5
m_4	Wikipedia/Yahoo individual mapping	OWL	o_4,o_5
m_5	Flickr/SWeMPs Mapping	OWL	o_6,swemps
m_6	Wikipedia/SWeMPs Mapping	OWL	o_5,swemps
m_7	Flickr/Wikipedia concept mapping	SWRL	o_6
m_8	Scenario Presentation Rules	SWRL	swemps

Service ID	Name	Input/Output
s_1	Restaurant Maps	SubjectMetadata(Gastronomy)/zyx:Image
s_2a	Flickr XML	zyx:Text/XMLDocument
s_2b	Flickr2RDF	XMLDocument/ResourceMetadata
s_3	Wikipedia Text extraction	Subject(w:Article)/zyx:Text
s_4	Wiki Travel metadata	Subject(w:TravelObject)/ResourceMetadata
s_5	Image transcoding	MediaType/MediaType
s_6	MPEG-7 mapping	XMLNamespace(mpeg7)/XMLNamespace(swemps)
s_7	Name extraction	Subject/zyx:Text

6.4.4 Thematic interest scenario type

SWeMPs should demonstrate the automated retrieval of content from multiple locations. We have two ways to acquire content:

- directly by reference in the SWeMPs conceptual model
- indirectly by invocation of a Web service which returns retrieved content

Note that the SWeMPs conceptual model does not need to directly reference content (Resources) but rather knowledge about it which is used for the semantic selection from a range of potential content.

The only `ResourceMetadata` available at the execution of the multimedia generation task is the annotation of the video with the modified version of MPEG-7 (which can be mapped to a SWeMPs representation). We can consider how a multimedia generation process finds other content (from other sources) automatically. Their selection is based on a common understanding of their description (based on SWeMPs) and their integration into the presentation through the relevant presentation rules.

Given the query within the system (using the abstract statement form):

```
k_sub     =  X
k_prop    =  {swemps:represents}
k_obj     =  y [uri: t:Sehenswuerdigkeit]
```

which means „what are all things which represent a specific instance of a touristic sight?", the use of ontology o_2 for a class in the query makes relevant the metadata file m_2 and the use of the namespace of ontology o_4 for the instance makes relevant the metadata files m_1, m_2 and m_4. The access to mappings between o_4 and o_5 made possible in m_4 makes metadata in the namespace o_5 relevant, and hence m_3. This in turn brings in o_3.

We want to look at the retrieval of content. This takes place through answering the query through a list of concepts, and for those concepts then determining resources that can be used to communicate them to a user. In this case the system query itself (at the stage of rule 4 in the process described in section 5.6) is the query for representations that normally takes place at rule 5. There are no `swemps:represents` statements in the current knowledge base so first the system checks for services for knowledge look-up and then – as no services are present – attempts to map to known ontologies and in this case finds mappings to both o_3 and o_6. So the system can also extract the following mappings to SWeMPs (using Venetian Pools as the example touristic sight) from the relevant sub-set of our metadata:

(mapping to o_3) – using s_6 to extract SWeMPs metadata from (modified) MPEG-7

```
…#vs0246456 [zyx:Video]
swemps:represents          http://en.wikipedia.org/wiki/Venetian_Pool
swemps:has-occurrence      http://...../video.mp4
zyx:start                   30000
zyx:duration                17000
```

(mapping to o_6) – using m_5 returns no results as there are no `foaf:topic` statements that can be mapped.

There is a second concept in the query to be tested for and that is the sight itself, taken from the Yahoo Travel namespace and classed in the Tourism namespace. In knowledge look-up we find two (composable) services which output `ResourceMetadata`: s_7 composed with s_2 can extract string keywords for a subject (based on key properties like dc:title) and use these keywords to acquire Flickr metadata, while s_4 will take a subject of type `w:TravelObject` to acquire SemanticWiki metadata.

(metadata by o_6) – acquired by s_2+s_7

```
…#image2625 [foaf:Image]
dc:title    "Venetian Pools"
foaf:topic  http://www.flickr.com/photos/tags/venetianpool
foaf:topic  http://www.flickr.com/photos/tags/miami
dc:source    http://photos22.flickr.com/2625_b64ad1_o.jpg
```

Here, the s_7 service extracts a name for the concept from the metadata m_2 and searches Flickr for images tagged with that name.

(metadata by o_5) – acquired by s_4

```
http://wikitravel.org/en/Venetian_Pool [w:TravelObject]
w:see       http://en.wikipedia.org/wiki/Venetian_Pool
```

Note how this is made possible through the mappings in the OWL file m_4, which establishes the equivalence between the individual of type `w:TravelObject` with the individual of type `t:Sehenswuerdigkeit` in the original query.

We now have metadata inserted into the conceptual model which is using the ontology o_6, which is mappable with m_5 (at the ontological level) and m_7 (at the instance level) to a SWeMPs representation:

```
…#image2625 [zyx:Image]
swemps:represents     http://en.wikipedia.org/wiki/Venetian_Pool
swemps:has-occurrence http://photos22.flickr.com/2625_b64ad1_o.jpg
```

Furthermore, metadata in the conceptual model using the ontology o_5 is mapped with m_6 which states that `swemps:represents` is the inverse property of `w:see`. The subject of type `w:Article` can be realized as a displayable resource through service s_3. This service extracts the first paragraph of text from the Wikipedia article. Finally, the service s_7 can take the subject itself (the instance of a touristic sight) and return a `zyx:Text` instance (i.e. a concept representing a textual resource). It queries for usual properties that textually describe the concept (`rdfs:label, dc:title, foaf:name` etc.), so in this case it finds a label in the metadata from m_2: "Venetian Pool".

To determine how this content is presented in relation to one another the system applies a number of presentation rules:

X before Y ← X zyx:start x Y zyx:start y x < y.	Y during X ← X swemps:is-of-type zyx:Video Y swemps:is-of-type (NOT zyx:Video[94]) X swemps:represents C Y swemps:represents C
Z appears-on-click Y; X pause-on-click Y; Y top-left-corner X ← Y swemps:is-of-type zyx:Text Y during X Z during X Y swemps:represents C C rdfs:label Text1 Y swemps:has-occurrence O O swemps:text Text1	R contains-h-layout Z ← Z appears-on-click Y

These rules state:

- The video segments are ordered sequentially according to their respective start times (we assume there are no overlapping segments)
- During any video segment, the other resources which represent the same concept are also included in the presentation

[94] Negation is an aspect of logics that is regularly debated when defining subsets of FOL. For example, the SWRL proposal does not include negation, though naturally the extension to first order logic SWRL FOL does. The W3C Rule Interchange Format working group (http://www.w3.org/2005/rules/wg) is working towards a standard for rules on the Semantic Web, and is expected to include support for negation-as-failure and classical negation (at least at one level of the rules language). Negation can be emulated by defining a class ("NotVideo") as disjoint to Video and checking for membership of this NotVideo class.

- Of those resources, the Text resource which is the label of the concept is displayed in the top left corner of the video. Whenever it is selected by the user, the video is paused and the other resources are displayed
- The display of the resources that appear when the label is clicked is laid out horizontally. The constraint `contains-h-layout` defines a ZyX complex media element that acts as a container for the other resources and enforces a certain layout upon them. The horizontal layout attempts to place all resources within the display screen next to each other horizontally, using a new vertical line when the display width is reached and adding interactive links and new screens if the resources are too many to display in a single screen. One can use different types of overflow strategy to solve this [Rutledge,2000].

As a result, the scenario realises a summary of a tourism video in which segments relating to sights in Yahoo Travel are selected and presented chronologically, and during their presentation the name of the respective sight is displayed in the top left corner of the video (Figure 6.7).

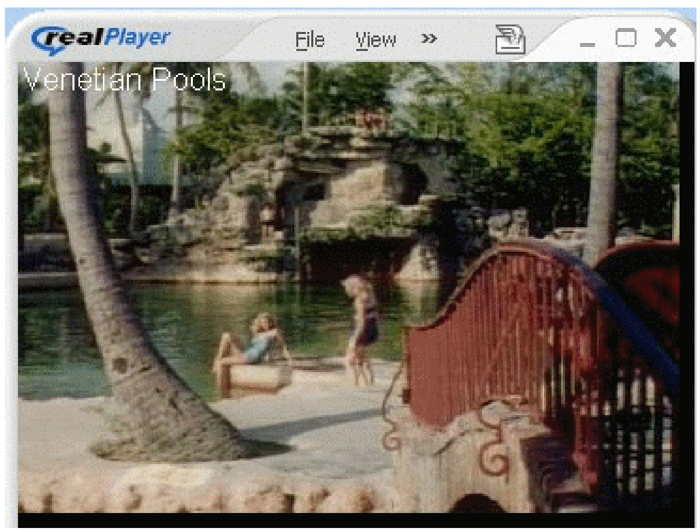

Figure 6.7 Scenario as SMIL Presentation in RealPlayer.

If the user interacts with this name (the form of the interaction would be set in the final formatting stage, dependent as it is on the target device and presentation format), the video is paused and media relating to the concept are presented: images extracted from Flickr and text from the related Wikipedia article (Figure 6.8).

To reiterate, we have described how the process has taken a single concept (here, the Venetian Pools in Coral Gables, Florida) and through the use of metadata and services it could find resources from four different sources (a tourist video, Flickr photos, Wikipedia text and a name from Yahoo Travel) which represent that concept, thus demonstrating **dynamic data integration**. We have also shown the **mediation between different knowledge representations**, at the ontological level both through a simple one-to-one mapping of classes and properties using OWL (m_5, m_6) as well as the less trivial generation of RDF statements from a XML data model (s_6), and at the instance level again both through a simple one-to-one equivalence (m_4) as well as the less trivial rules-based mapping (m_7).

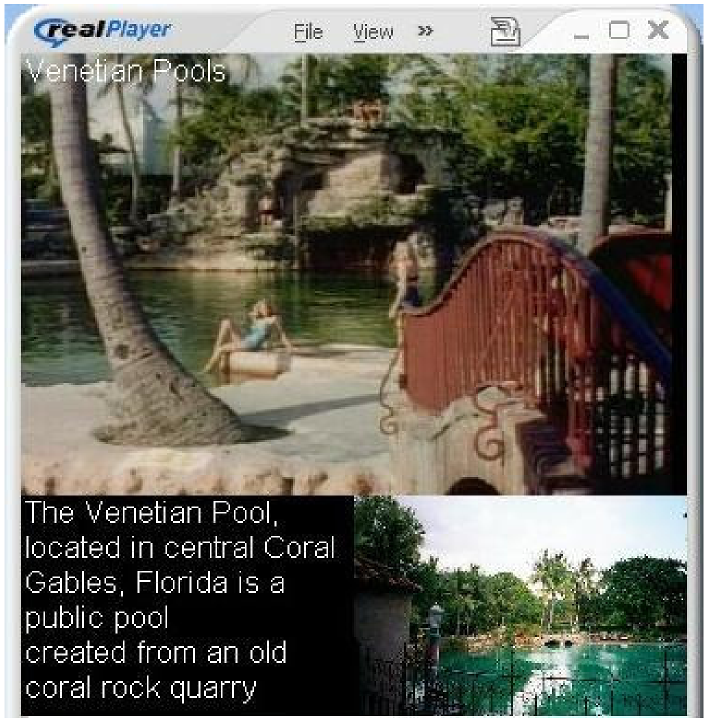

Figure 6.8 Selecting the label displays additional content from the Web (Wikipedia, Flickr)

In the scenario, initially only metadata for the tourism video and the Yahoo Travel-scraped knowledge about restaurants and sights is available. The Flickr and Wikipedia information demonstrates **dynamic knowledge retrieval**, where adding services to the conceptual model of the multimedia generation process that enable the retrieval of metadata from these sites combined with mappings from the metadata formats they produce and instances that they use has permitted the system to acquire and integrate additional resources into the presentation. These additional resources are linked to the segments of the video that deal with the same concepts, demonstrating **coherent presentation**.

6.4.5 Focused touristic scenario type

We turn to another scenario using the same multimedia generation model but generating a different presentation on the basis of a different initial query, which could be generated from a user interface which presents the user with a menu of different cuisine options:

```
k_sub     =  X [uri: r:Restaurant]
k_prop    =  r:typeOfCuisine
k_obj     =  y [uri: r:Cuisine]
```

This query draws on the Gastronomy ontology o_1, and as the Cuisine instance also exists within the namespace of o_1 the metadata m_1 is added to the conceptual model. In resolving the query for a particular cuisine, a number of restaurants match from the Yahoo Travel metadata scraped from the website. The system must now resolve resources that represent those restaurants for presentation to the user.

The restaurant concept exists in the Yahoo Travel namespace. While the Flickr and Wikipedia services could be used in this case as well, matches are not found for these specific restaurants. However service s_1 composes a means to extract a Yahoo Map from the restaurant description, by generating an URL which conforms to the Map Image API from the restaurant RDF metadata using the street address and city properties.

Another means of resource extraction is the service s_7, provided it is able to identify the properties `r:name` and `r:address` as identifiers for text, either by being extended to recognise this ontology or in that the ontology marks the properties as such by making them equivalent or subclasses of a property such as `rdfs:label`.

Again we turn to the presentation rules to order these resources in a multimedia presentation, drawing upon their subject and resource metadata:

<table>
<tr>
<td rowspan="2">X title Y; Y subtitle X ←
X swemps:is-of-type zyx:Text
Y swemps:is-of-type zyx:Text
X swemps:represents C
Y swemps:represents C
C r:name Text1
X swemps:has-occurrence O1
O1 swemps:text Text1
C r:address Text2
Y swemps:has-occurrence O2
O2 swemps:text Text2</td>
<td>X bottom-right-of Y ←
X swemps:is-of-type zyx:Text
Y swemps:is-of-type zyx:Image
X swemps:represents C
Y swemps:represents C</td>
</tr>
<tr>
<td>X before Y ←
X swemps:represents C1
Y swemps:represents C2
NOT (Y before X)</td>
</tr>
</table>

Here the text is constrained as being either a “title” or a “subtitle” of another text, through which the relative style of the text is defined and its general spatial positioning is specified. Both text resources are constrained to being placed in the bottom right corner of the image that they relate to conceptually. Finally, the resources for each distinct concept (a restaurant) are ordered temporally through a “before” rule. Note the use of the negation in the rule to ensure that once a set of resources are placed temporally before another set, the system can not make a contradictory inference.

The result of this particular query is a slideshow effect which could be combined with the video-based presentation of the previous subscenario. An additional option during video playback provides users with access to the restaurant guide, leading to a pause in the programme and the user selecting a cuisine leads to the formulation of the query given in this section and its submission to the information service. The result is the slideshow in which the Yahoo maps are displayed together with the restaurant name and address (Figure 6.9).

Again, there is the combination of resources from different sources (Yahoo Maps and Yahoo Travel), and their presentation based on their relationships (an Image is titled and subtitled by resources representing a restaurant name and address, respectively). Presentation of maps with labelled objects is of course already done, but without the use of semantic information. With the underlying semantic approach here, we note that this sort of scenario not only is initiated by a semantic query (and hence finds its content through reasoning over available knowledge rather than the user needing to select manually the desired content) but also can be extended by the addition of further content sources and services. Hence other annotated media relating to the restaurant, e.g. wiki-based open content about the restaurant such as visitor reviews, could be found on the Semantic Web, processed automatically (with the help of ontology mediation) and integrated into the final presentation through an appropriate presentation rule. We will take this scenario a step further to illustrate this important principle of the SWeMPs approach, introducing also the idea of adaptation to context, the remaining requirement for which we are evaluating the SWeMPs approach.

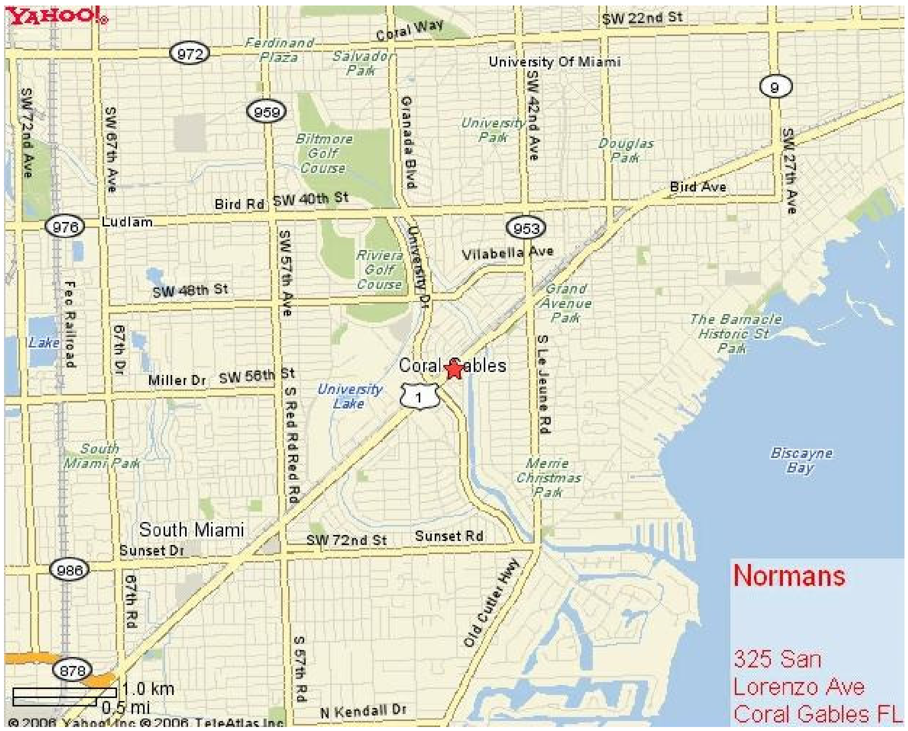

Figure 6.9 Yahoo map with restaurant information

6.4.6 Immediate viewing scenario type

We add to this scenario the context that the user is using a mobile device and is located presently within the city about which the tourism video is relating. The user is hence not only interested in finding a restaurant, and that it serves a cuisine which he or she likes, but also that the restaurant is located close to his or her present location.

Note that here the user should be presented with the relevant information with the minimum of interaction – e.g. the multimedia generation process is perhaps launched by a single click representing “Give me restaurants I like close to here”. Both the user’s preferred cuisine and current co-ordinates (as provided by GPS for example) are inserted into the multimedia generation process by adding them to the presentation constraints sent as input to the system, along with the device characteristics.

Additionally, we introduce a service into the conceptual model which takes as input two locations in the form (latitude,longitude) and returns the distance between those two locations in terms of the proximity vocabulary (using a `prox:distance-from-user` property). The `r:address` and `r:city` properties are used with a geolocation service to derive their (latitude, longitude) values and hence the Restaurant metadata can be made compatible with the proximity service in order to compute the distance between user and restaurant.

Hence this last scenario introduces a few more ontologies, a mapping file and a service into the conceptual model (the other metadata is placed into the presentation constraints passed as input to the generation process, rather than into the conceptual model):

Ontology ID	Title
o_7	FOAF User Profile
o_8	GeoRDF
o_9	Proximity
o_10	CC/PPx Device Description

Metadata ID	Title	Ontology	Namespace
m_9	CC/PPx to SWeMPs mapping	OWL	o_10

Service ID	Name	Input/Output
s_8a	Geolocation Service	SubjectMetadata (Gastronomy)/(GeoRDF)
s_8b	Proximity Service	SubjectMetadata (FOAF)+(GeoRDF)/(Proximity)

Here the query expressed to the system is a join between three queries

```
1. X [uri: r:Restaurant] {r:typeOfCuisine} Y [uri:r:Cuisine]
2. Z [uri: foaf:Person]  {foaf:likes}      Y [uri:r:Cuisine]
3. X [uri: r:Restaurant] {prox:distance-from-user}
       d [xsd:decimal] <= 2.0
```

In other words, select those restaurants whose type of cuisine is liked by the current user and that are in close proximity (2km or less). Note that in this scenario not only user but also device and the proximity service play a role in the selection and presentation of the resources. Firstly, the addition of the selection according to the user's preferences filters the available restaurants. As in the previous scenario, the system can retrieve for each restaurant an Image (Yahoo Map) as well as some text (name and address). However, the mobile device has a smaller display so the map must be scaled down and the end format is SVG rather than SMIL. Finally, to test for the third query we find no metadata using the proximity vocabulary defined in o_9 so services are examined for the production of metadata using this ontology. The proximity service s_8 is selected, and returns `prox:distance-from-user` statements for the selected restaurants.

The text resources are placed spatially and styled according to the same "title", "subtitle" and "bottom-right-of" rules from the previous scenario. The only other rule that need apply in this scenario is a temporal positioning of the restaurants based on their relative distance from the user, replacing the previous scenario's looser temporal ordering rule. The new rule checks the `prox:distance-from-user` of the restaurants and places those whose distance from the user is a lower value (i.e.

closer) earlier temporally in the presentation. Thus the selected resources are organized in a presentation according to the following presentation rules:

X title Y; Y subtitle X; X bottom-right-of Y (as above)	Y before X ← Y subtitle S X title T Y swemps:represents C1 C1 prox:distance-from-user Value1 X swemps:represents C2 C2 prox:distance-from-user Value2 C1 < C2

This scenario demonstrates not only the re-use of presentation rules and how a presentation can be extended to provide new functionality through the identification of relevant ontologies, metadata and services, but also how the requirement of **adaptation to context** is met, here through three distinct contexts:

- User preferences
- User location
- Device characteristics

Figure 6.10 shows the scenario result, which in comparison to Figure 6.9 is SVG rather than SMIL and has a smaller window size.

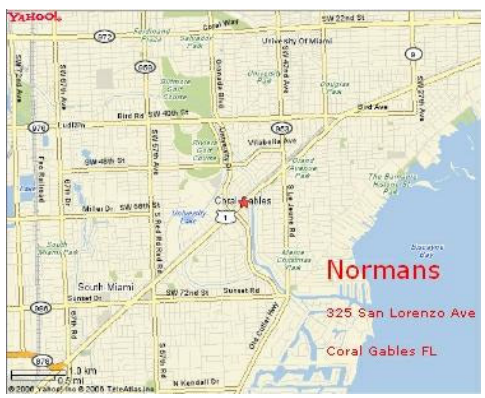

Figure 6.10 Yahoo map with restaurant information
(based on proximity information and delivered to a mobile device)

6.4.7 Summary

The second scenario has dealt with real world data that could be extracted from the Web and integrated through the SWeMPs approach into a multimedia presentation generation process. The table below provides an overview of the means by which this scenario was realized (numbered by the respective subscenario given in this section).

		ONTOLOGY (6+4)	METADATA (8+1)
Knowledge		1/2- Gastronomy and tourism found on Web; Yahoo/Wikipedia schemes derived from website. 3 – Extended with contextual ontologies	1/2- Restaurants and sights scraped from Yahoo (using Piggy Bank)
Resources	1/2- Video open source and online; Flickr photos/Wiki text reused	1/2- MPEG-7 and Flickr as metadata schemes	1/2- MPEG-7 annotation derived from manual usage of IBM tool
Services	1/2- 7 services coded specifically, incl. to support existing services with SWeMPs. 3- 1 additional service coded specifically	1/2- OWL/SWRL as mapping language; otherwise OWL-S	1/2- two ontology mappings and two instance mappings using OWL or SWRL; other services use SWeMPs properties or OWL-S IOPE descriptions 3- additional OWL based mapping and additional OWL-S description
Presentation	1/2- Created by hand (4 rules in first subscenario, 3 rules in second subscenario) 3- All but one rule reused; one rule rewritten	1/2- SWRL used	None required

The realization of this scenario has demonstrated that:

- Even in the absence of much Semantic Web data in the present Web, the conditions are in place for extracting Semantic Web-compliant data from non-semantic sources. Hence the Semantic Web basis of SWeMPs is not invalidated by the relative non-existence of Semantic Web data at the present time.
- There are some ontologies already discoverable on the Web which can be used to formally annotate Web content and tools and methodologies being established to generate this annotation, both of structured text (XML/XHTML) and of non-textual resources (e.g. the aceMedia work described in section 2.1.1). Hence ongoing parallel research feeds into the SWeMPs approach.
- The SWeMPs approach can scale beyond the small data sets used to illustrate its operation in the previous family tree scenario. Its efficient operation may be limited by current performance limitations in Semantic Web reasoners and rule engines, however the use of Prolog as an internal knowledge representation allows us to already reason efficiently according to a well established and researched logical formalism: we have acknowledged

some differences in that formalism compared to Semantic Web approaches (section 5.2) yet found that the scenario could be realized within the common subset termed DLP (one open issue was negation, see the comment in 6.4.4).
- There is much potential for the re-use of appropriate sources of metadata, resource annotation, Web services and presentation rules, hence leading to the realization that this approach benefits from its own use, i.e. as libraries of appropriate content could be built up in the Semantic Web the policy of re-use would support the simplified creation of Intelligent Information Services by allowing developers to reference existing sources in the SWeMPs conceptual model.

In terms of the requirements we have detailed for the SWeMPs system, we summarize how each subscenario was found to meet particular requirements in the table below:

Requirement from section 1.4	Thematic interest (6.4.4)	Focused touristic (6.4.5)	Immediate viewing (6.4.6)
1 – data integration	X	X	X
2 – representation mediation	X		X
3 – adaptation to context			X
4 – dynamic knowledge retrieval	X		X
5 – coherent presentation	X	X	X

As the table shows, the realisation of this scenario achieves our aim of demonstrating that SWeMPs meets all of the requirements of a next generation IMMPS.

6.5 Evaluating the domain independence of the scenarios

Through the scenarios presented here, in their implementation and realization, we have been able to demonstrate the SWeMPs framework in a concrete fashion, drawing on existing (or possible-to-exist) sources of knowledge, content and presentation rules. In the first scenario, we used a simpler data set to allow for a concrete description of the multimedia generation process and explain its result. In the second scenario, we chose an emerging multimedia delivery channel (interactive television) which would clearly benefit from automatic and adaptive multimedia generation and demonstrated the use of SWeMPs to support a tourist program. In this case, we selected some possible sub-scenarios to provide a proof that the approach is capable of meeting the requirements that have been specified. However, we added an additional requirement in section 6.2: domain independence. Both scenarios could prove that the SWeMPs approach works when generating family trees or interactive tourism programs, and nothing else. Hence it is important to be able to show if the approach outlined in the thesis can be broadened to any domain, apart from the two scenarios chosen here for the purposes of the evaluation.

While the first scenario was based around the single domain of genealogy, the second scenario drew from the domains of tourism and gastronomy. In both, we were able to identify the constituent domain-specific content and categorize it as belonging to the concept of knowledge, resource, service or presentation. Furthermore, besides the content itself, we could identify the metadata (content description) and the ontologies used to guide the system in selecting and using the content. We see that the SWeMPs conceptual model contains all of these concepts and hence all domain-specific content can be abstracted as instances expressed in the knowledge base provided to the SWeMPs system when generating a multimedia presentation (hence we validate the division illustrated in Figure 6.2). Given that the underlying representations of that content are consistent regardless of which domain is being considered (e.g. all ontologies will be in RDF/OWL), the SWeMPs multimedia generation process is able to produce a multimedia presentation independently of the domain. Hence we can conclude that the approach would be exactly as valid if working with medical content, or e-commerce data, or some other domain.

6.6 Conclusion

In the evaluation, we have used two scenarios in order to demonstrate not only the viability of the approach proposed in this thesis but also its realisation in the form of the SWeMPs framework. We have shown how intelligent information services can be authored by re-using existing data (or preparing suitable data in advance, often through extraction from existing data), adapted and dynamically changed by reasoning on metadata and can generate presentations based on the higher level conceptual relationships that are known to exist between (the concepts represented by) the selected media resources. In comparison to other approaches to IMMPS development which are not tightly coupled to the Semantic Web vision, we have shown how SWeMPs meets requirements for a future automatable and adaptive intelligent multimedia generation framework which demonstrates domain independence. In the following chapter we consider the contributions of this work and conclude by looking at possible future developments and research directions.