Multimed Tools Appl
https://doi.org/10.1007/s11042-018-6080-8

Spatial reasoning about multimedia document for a profile based adaptation

Combining distances, directions and topologies

Yehya Belhadad[1] · **Allaoua Refoufi**[1] · **Philippe Roose**[2]

Received: 23 November 2017 / Revised: 19 April 2018 / Accepted: 30 April 2018

Abstract The continuous evolution of smart devices has led to serious limitations in multimedia applications. The multimedia graphic design and animation and the increased use of rich and complex multimedia content on the web or other media have all created a need to diversify the accessibility of the content. Therefore several techniques are used today to design a universally accessible content. The main techniques that are still used to maintain accessibility is to create two parallel streams of design and development of the same content. Thus, the continuous evolution will certainly lead to create other streams. For this, the automatic reasoning on multimedia to allow a computer to modify the design according to different variables, devices capabilities, user status and context to provide personalized adapted content. In this paper, we propose an abstract document model called XMS short for XML Multimedia Specification; it describes the composition of an original multimedia document and can be extended to any document type. We present how we may use spatial information present in this document to adapt and modify the original document. We mainly focus on the spatial aspect of a web document, a combination of RCC8, qualitative distances, and directions are used to describe the layout of a set of objects. The level of granularity of the definition of the objects defines the level of details that will be processed by our PROLOG based inference system, simplified versions of algorithms from the inference system and how it works on the spatial dimension of the document are shown. In the end samples of how spatial relations manipulation algorithms work are illustrated.

✉ Yehya Belhadad
belhadad.yehya@univ-setif.dz

Allaoua Refoufi
allaoua.refoufi@univ-setif.dz

Philippe Roose
Philippe.Roose@iutbayonne.univ-pau.fr

[1] University of Ferhat Abbes Setif 1, Setif 19000, Algeria

[2] LIUPPA, IUT de Bayonne, 2 Allee du Parc Montaury, Anglet 64600, France

Published online: 25 May 2018

Keywords Content adaptation · Multimedia document · Spatial reasoning · PROLOG · Inference

1 Introduction

Recently and in the late technological advancements, new and emerging technologies are seen every day on a continuous basis. In each of these advancements, the difficulty of creating new uses for these technologies becomes harder than the previous level. Mainly because of the growing speed of creating new devices that cross the edges and limitations of the previous ones. Multimedia devices are no exception to this, every day we get a step further from the earlier technologies; however, the software industry improvements and evolutions are usually slower than the hardware industry. In this wide evolutionary environment, multimedia design and publishing services struggle to create a universally accessible multimedia (UMA: Universal multimedia access) [7, 40, 41]. Several works tackle this issue using different approaches; the main used approach is to maintain alternatives such as for mobile devices [1, 3, 31, 36, 43].

Today, we are continually approaching the creation of new devices such as virtual reality technologies. Not only mobile devices in these days are touched by considered an important multimedia terminal. Web browsing still not used as part of the virtual reality industry, this may lead shortly to the need not only to adapt web multimedia to limited devices. In the first years when adaptation was proposed as a problem [32] only mobile devices were considered. Shortly adaptation would be needed to adapt multimedia documents designed for conventional computers to multimedia and web content for more advanced devices. For this, based on device characteristics or user context automatic reasoning and adaptation of multimedia is needed to provide customized multimedia content to users. In this paper, we show how we can use spatial reasoning methods to adapt a given original document to a new version according to a target output.

1.1 Multimedia document adaptation

Before we get into multimedia document adaptation, first we need to give the multimedia document a definition.

Definition 1 According to Ellis [11] multimedia is defined as “a computer-based product that enhances the communication of information by combining two or more of the following: text, graphic art, sound, animation, video, or interactivity”.

Another definition of multimedia documents is given by Li et al. [30].

Definition 2 The authors define the term ‘multimedia’ as “a name given to a collection of data of multiple types, which include not only “traditional multimedia” such as images and videos, but also emerging media such as 3D graphics (like VRML objects) and Web animations (like Flash animations)”.

Based on the previous definitions (Definition 1, Definition 2) one can understand that a multimedia document can contain any form of data that can be interpreted, presented by the terminal device and perceived by the final user. Therefore, to interpret and present such data, we will face the diversity of possible situations, where users can view or use these multimedia documents; from a basic two-dimensional static or dynamic output to three-dimensional static or dynamic content see multi-sensorial multimedia technologies [23, 42]. These situations vary on several levels, on a hardware level, a software level, and a user level which comprise in the context and preferences or even his incapability [10, 21, 24, 33, 35]. Executing a possible presentation involve interpreting the information within the document. The several usage situations may interfere with the interpretation sometimes by constraining the execution, and sometimes by the misinterpretation of the structure of the document resulted from the misuse of the coding standards by software or hardware developers.

All these variables generate a need to create a solution for adapting such documents to fit in all these situations. Several techniques have been proposed; they can be classified according to [29] into these main approaches: an approach based on alternatives specification [4–6, 19]. Another based on transformation rules [9, 12, 20], and another approach based on flexible document model [18, 37].

1.1.1 Adaptation based on alternatives specification

In the proposed approach of adaptation solutions, the authors have to specify the appropriate alternatives in case of incompatibility of the default multimedia document with either the terminal or the user context. This solution requires that the author knows in advance all possible situations where and when their default document or some components cannot be presented properly. They can specify the triggering constraints that allow the adaptation engine to find a suitable version, or simply define the target profile properties such as the screen size or the operating system.

1.1.2 Adaptation based on transformations rules

Instead of defining a whole alternative based on constraints or properties. In the proposed approach, the author has to specify rules for the adaptation engine to perform a set of transformations on the default document or a part of it, just like the alternatives specification the author has to know all the possible situations that need adaptation in advance.

1.1.3 Adaptation based on a flexible document model

Instead of forcing authors to predict all possible situations where their documents should be executed, in solutions based on this approach, the author specifies a set of decomposed or semi-composed objects then the adaptation engine is the responsible for composing the final presentation based on these objects [12]. Doing this, the authors have not to know the possible situations of execution of the presentation. Meanwhile, they have no control over the final presentation.

1.2 Adaptation approaches factors

Previously, a glimpse view of adaptation approaches has been provided. Each of the presented techniques has a proper application, Table 1 shows some differences based on the following five factors are summarized:

- The first factor is about the level of abstraction of the adaptation approach, a highly abstract one would exclude the author from the process of adaptation, this means that the author should not care about the adapted version of his document. Instead, the developers of the approach are the one who takes full responsibility for the adaptation part; this means that the approach is entirely automatic and the authors have no control over it.

Table 1 Adaptation factors for used adaptation technique

Adaptation technique	Level of abstraction	Author workload	Level of separation	Level of flexibility	Author control
Based on alternatives	0	2	0	0	2
Based on transformation rules	1	1	–	1	0
Based on flexible document model	2	0	0	2	0
Ideal approach	2	0	2	2	2
Our approach	1-2	1-2	2	1-2	1-2

- The second factor is about the author workload; a good approach should minimize the workload of the author so he would concentrate more on the content rather than the process of adaptation. Most of the existing approaches give either full control with lots of work to the authors; or minimize the work of the authors while giving him no control over the adaptation process.
- One of the important factors here is the level of separation. By separation, we mean separating the original content and adaptation that might be in the description of alternatives or the layout and formatting of the content of a multimedia document, as we can see in Table 1, the usually used approaches do not consider this. Existing approaches are based on the content meanwhile, in this work, the adaptation parameters are either expressed with the content explicitly in the form of rules or something else, or in other cases are automatically extracted from within the content based on the properties of objects or the document itself.
- Another important factor is the level of flexibility which defines how much flexible the approach can be. A flexible approach should allow maximum flexibility over time without the intervention of the author, given a document created to be widely adaptable, with the appearance of new technologies the author of this document according to a flexible approach should not worry about the continuous adaptability.
- An underestimated factor in multimedia document adaptation approaches is the control of the author; authors need to have a maximum control while maintaining the rest of other factors at optimality. Usually, the adaptation approaches try to minimize the author workload and not think about the control of the author; the approach developers have no authority over the content of the multimedia document that has been created by an author. Some authors might consider consenting the developers such authority while others simply want the full control over their document, which leads them to only one option is to create a solution based on alternatives; where this approach is the one that puts the more workload on authors.

The next qualitative values are considered for comparison between the approaches:

- 0 = low value
- 1 = mediocre value
- 2 = high value
- - = depends on the implementation.
- 1-2 = optional depends on the authors' choice.

Table 1, the ideal approach is used as a benchmark; it should maximize all the factors while minimizing the author workload. It is clear that each approach has a strength in one or two factors. However, it may not be possible to create one ideal solution that works for every situation while maintaining other factors at optimality. To develop such solution, a priority of consideration for these factors is needed. The most important factor considered in our approach is that of the author control because he is the one who knows better the content of his multimedia document and how it should be adapted to different situations.

The proposed approach in the table sets the goals we want to reach; to maximize the author control over his multimedia document a flexibility on the other factors is a must. This way allows us to maintain a balance in all the factors we proposed, the author has to consent us to have control over his content, but if he chooses to have the full control, then we should allow him to use our solution and yet maintain the optimality of other factors.

To allow such flexibility while maintaining the authors' control, our approach is based on an abstract document. This document (XMS) is used in the processing model instead of the original document; a basic version of the abstract document might be generated from the original one. Advanced options can be defined by the author to give him the control he wishes over the document either by using alternatives, restricting or directing transformations.

1.3 Paper organisation

Section 1 of this paper provides a glimpse view over the multimedia document adaptation, approaches, and factors. Section 2 proposes the novel multimedia document processing model XMS. Section 3 provides how the abstract document model XMS allows creating an abstract spatial specification using a combination of qualitative distances, directions, and topologies.

Section 4 explains how and abstract spatial specification can be manipulated and provides the basic algorithms. Section 5 presents the main algorithm that uses algorithms from Section 4 to adapt a given abstract spatial specification.

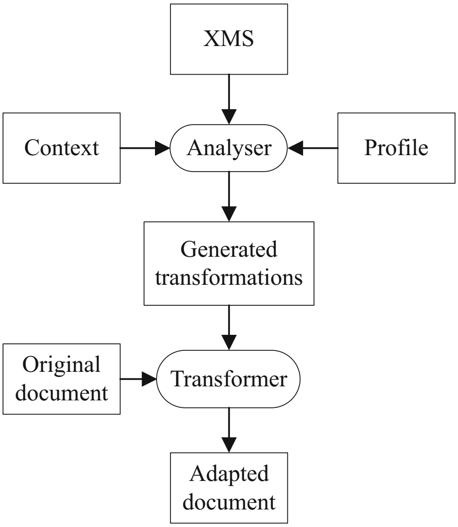
Section 6 presents samples how the manipulation algorithms work on an example simple abstract web page layout, compares the work with other similar works and provide an extensive comparison with solutions that tackle the adaptation of multimedia documents. Finally Section 7 concludes the paper.

2 A novel multimedia document processing model

Our processing model takes as input a description of the original document we call XMS (XML Multimedia Specification); this file written in XML contains an abstract version of the original multimedia document. The fact that the processing model takes as input an XMS document instead of the original document helps us to maintain some factors at optimality (Separation, abstraction, flexibility). The components of this abstract document are the key elements for all other factors we set as a goal (author workload, author control, abstraction, flexibility).

In the web in general or any other multimedia delivery system, all begins with a user requesting a document on a client device; the multimedia server then delivers the requested document. In this processing model (see Fig. 1), the author creates or generates an abstract version for his multimedia document. After a request, the server takes into account a set of parameters and properties that describe the state of the requesting user device or even the user preferences and context, and instead of processing the document directly the system takes as an input the abstract document XMS. It allows the processing engine to decide if an adaptation is needed, and eventually, it delivers the original document as it is or generate a set of transformation and passes it to the transformation engine that uses the original document and a set of transformations to produce the adapted version.

Fig. 1 Approach architecture



2.1 XMS: an abstract document model

The core of the XMS document model is based on the following definition of a multimedia document. A multimedia document D is defined as: "$D = \langle O, S \rangle$ where O a set of N media objects and S a set of M specification". Each specification concerns one of the dimensions of the document; which can be either spatial, temporal, hypermedia or logical dimensions [12], to make the XMS more generic, the logical dimension is considered as a part of the original document and is not defined explicitly in the abstract document. This also helps us to maintain a level of abstraction to give a space for the creativity of the author.

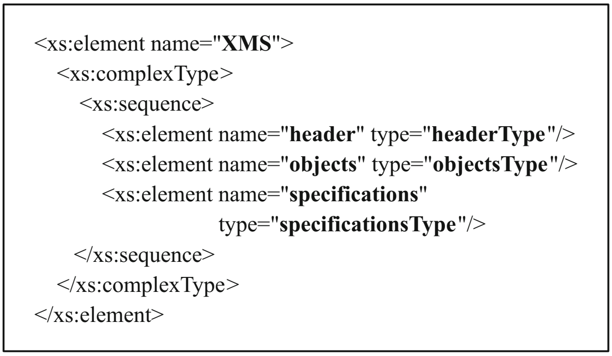
$O = \{O_i | 0 < i \leq N, i \text{ and } N \in Z\}$ a set of N media objects O_i, an object can be another multimedia document in complex documents such as in web or simply a basic media element.

Definition 3 A media element is the basic element used by the author to compose his final multimedia document, it can be a text, images, videos or any other medias.

Definition 4 An object or an abstract object in XMS to describe the composition of a multimedia document is a set of at least one basic media element; it is an indecomposable part of the composition. An object can be defined as an essential element of a multimedia composition created by an author. Given a document D composed of N media elements, the number of objects is M where $M \leq N$.

Definition 5 Specification granularity: the number of objects in a specification by the number of basic elements defines the level of granularity of a specification, a granularity equals to one is the most detailed specification of a multimedia composition. The granularity can be part of the interval]0, 1], the bigger the granularity is the more detailed the specification we get.

$S = \{S_k | 0 < k \leq M, k \text{ and} M \in Z\}$ a set of M specification, each one is a couple of two sets $S_k = \langle I, R \rangle$ where:



```
<xs:element name="XMS">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="header" type="headerType"/>
         <xs:element name="objects" type="objectsType"/>
         <xs:element name="specifications"
                        type="specificationsType"/>
     </xs:sequence>
   </xs:complexType>
</xs:element>
```

Fig. 2 XMS main components in XSD

- $I = \{i \in Z | O_i \in O, 0 < i \leq N\}$ the set of identifiers of objects concerned in the specification, it may or may not concern all the object in O.
- $R = \{R_l | R_l = \langle i, j, C_n \rangle, i, j \in I, C_n \in C\}$ is the set of binary relations between these objects. C is a set of constraints that are specified usually using logic based on the dimension of the specification such as (RCC8, Qualitative distances ...etc.).
- Usually, the relations in R are exclusive, so if a constraint is not met, the specification is not either. Therefore, we get another different composition rather than the original intention of the author.
- The specification of multimedia documents can be implicit in the definition of the document; in XMS, a basic specification can be generated from the original document itself. Another possibility for this is to make the authors provide a specification explicitly in it, or giving a separate file to generate the final specification as an XMS document.

The previous definition is for a multimedia document; meanwhile, an abstract multimedia document does not contain the set of object. Instead, the abstract document contains only the set of identifiers referring to the actual objects in the original document. The Fig. 3 shows an abstract UML design of an XMS document.

2.2 XMS main components

XMS is composed of three main elements, a header containing the main declarations, the objects elements defining and referencing the objects of the composition and the specifications element that contains information about the dimensions of the document. The main XMS elements written in XSD schema are shown in Fig. 2.

3 Abstract spatial specification using XMS

To define a spatial composition of a multimedia document or its layout, a look at the used formalisms to describe spatial relations is needed. An abstract spatial specification needs to define a set of qualitative composite relations using both topological, and directional relations refined with qualitative distances [13, 17, 38].

The next elements defines three different spatial information used in XMS to define an abstract spatial composition (Fig. 3).

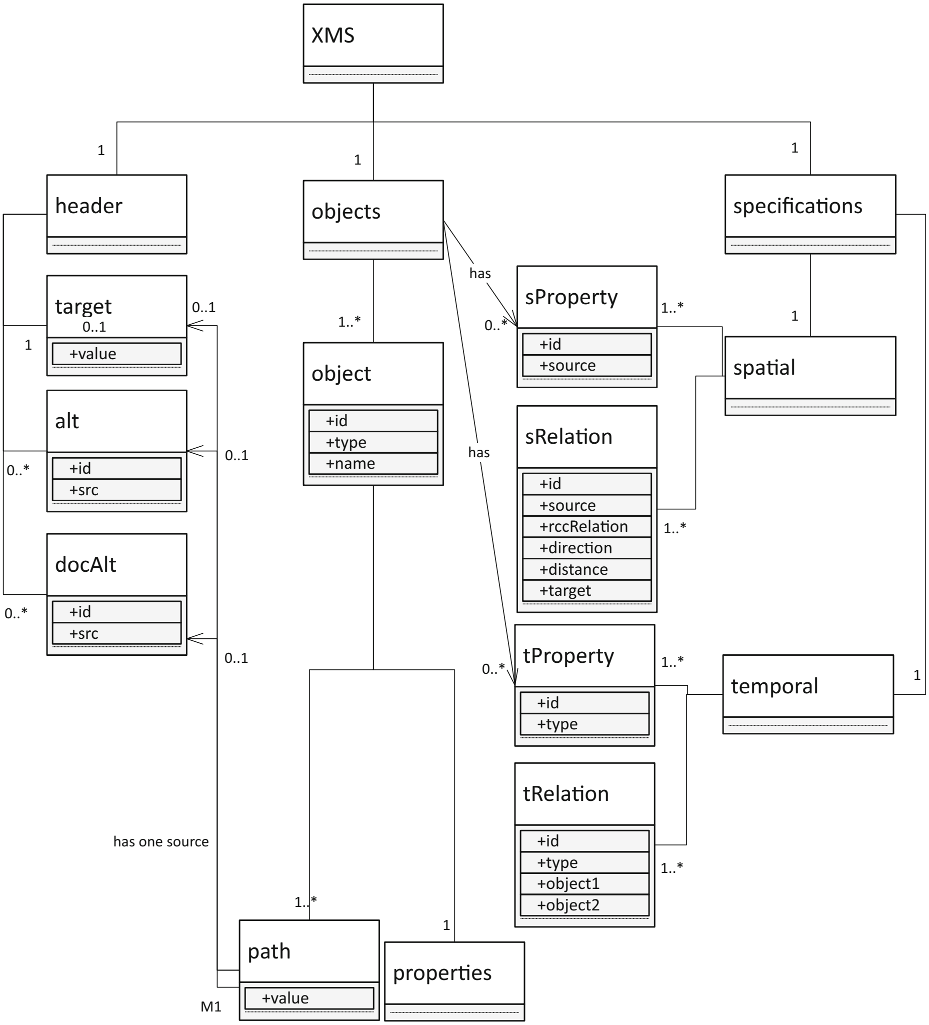


Fig. 3 XMS abstract UML design

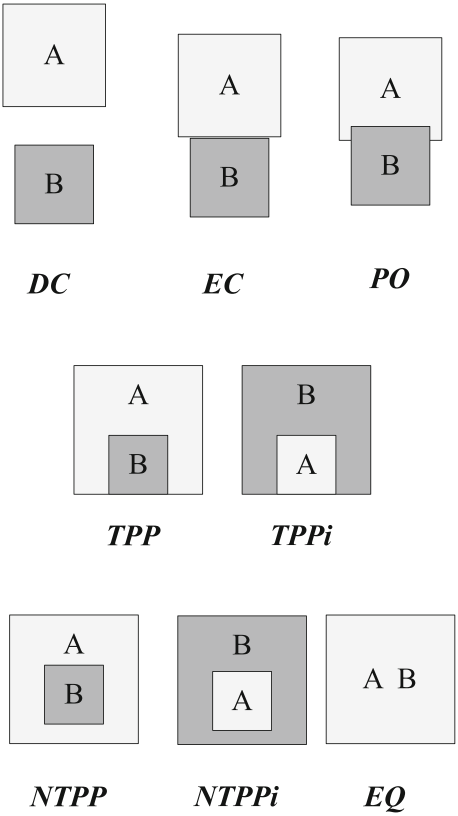
3.1 Qualitative topological specification

To represent topological information in XMS, the well-known RCC8 [16] are used (Fig. 4). This paradigm covers eight basic topological relations in a two dimensional space. These can be changed according to a conceptual neighborhood graph (Fig. 14) in order to adapt a composition.

Each basic relation in RCC8 describes the topology between two regions A and B.

- DC (is disconnected from): the two regions are disconnected the one from the other, the region A and B does not contain one another nor the borders are in contact.
- EC (is externally connected): the two regions are not included the one in the other but the external borders of region A are in contact with those of region B.

Fig. 4 The eight topological relations in RCC8

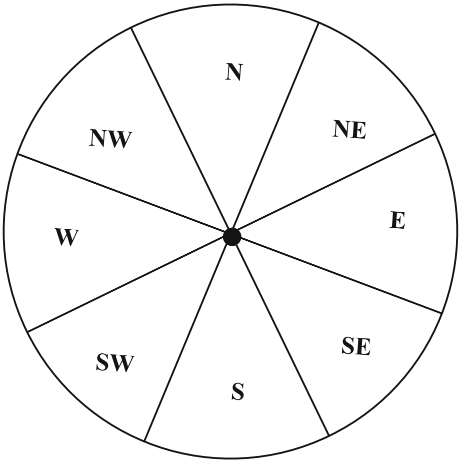


- PO (partially overlaps): the two regions are overlapped by one another; the region B overlaps the region A.
- TPP (tangential proper part of): $A\ TPP\ B$ the two regions are included one in another in this case A is included in B, and the borders of region A are in contact with the borders of region B.
- $TPPi$ (tangential proper part of inversed): $A\ TPPi\ B$ same as $A\ TPP\ B$ but in this case B is included in A, this is the equivalent of $B\ TPP\ A$.
- $NTPP$ (nontangential proper part of): $A\ NTPP\ B$, the region A is included in the region B and the external borders are not in contact.
- $NTPPi$ (nontangential proper part of inversed): $A\ NTPPi\ B$, same as $A\ NTPP\ B$ but in this case the region B is included in the region A, this is the equivalent of: B $NTTP\ A$.
- EQ (equals): $A\ EQ\ B$ or $B\ EQ\ A$, the two regions have the same position and the same size and borders.

3.2 Qualitative directional specification

Same with the topological description of a spatial composition, in order to define an accurate position of objects in a spatial composition we need to add information about the direction of an object according to a reference object Fig. 5. The direction is defined by the direction

Fig. 5 The qualitative directional relations



of the center weight of each objects. According to the previous figure, we count nine distinct directional relations. A directional relation between two objects A and B is described as follows: A Relation B, the nine relations are explained in the following items:

- **None** ($A\ NON\ B$): no direction, same as $B\ NON\ A$. In the case the center weight of each objects are on the same position. This is a neutral zone and simply describes the concept of here.
- **North** ($A\ N\ B$): the center weight of the object A is on the north of the center weight of object B.
- **North-east** ($A\ NE\ B$): the center weight of the object A is on the north-east of the center weight of object B.
- **East** ($A\ E\ B$): the center weight of the object A is on the north-east of the center weight of object B.
- **South-east** ($A\ SE\ B$): the center weight of the object A is on the south-east of the center weight of object B.
- **South** ($A\ S\ B$): the center weight of the object A is on the south of the center weight of the center weight of object B.
- **South-west** ($A\ SW\ B$): the center weight of the object A is on the south west of the center weight of object B.
- **West** ($A\ W\ B$): the center weight of the object A is on the west of the center weight of object B.
- **North-west** ($A\ NW\ B$): the center weight of the object A is on the north west of the center weight of object B.

3.3 Qualitative distance specification

To define the spatial composition, information about the space between two objects needs to be considered in the description of the spatial relation in a specification. Here we describe the qualitative distance specification of a spatial relation between two objects Fig. 6.

There are many paradigms when it comes to qualitative spatial specification, and qualitative distances are no different. The distance in this work is described as the distance from the center weight of the two objects A and B, and not the distance between the borders of

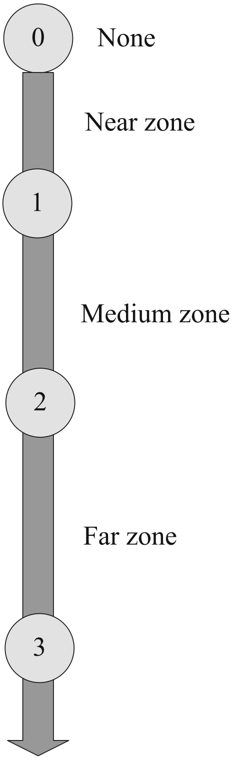


Fig. 6 The qualitative distance relations

the objects. To maintain a comprehensive specification, the authors have to define a set of intervals where the qualitative values might be, the next figure shows a set of four basic qualitative distances.

Four different qualitative distances are used in XMS, the first is the *none* where no distance is present between the two center weights of the two objects A and B, and are on the point 0 in the next figure. There are three different intervals where the rest of the qualitative distances can vary. The second qualitative distance is the near distance where the center weight of the object A is on the point 0, and the center weight of the object B vary in the interval $]0, 1]$. When the center weight of the object B surpasses the point 1; the qualitative distance then takes a medium value. In the medium qualitative distance, the center weight of object A still at point 0 and the center weight of the object B vary in the interval $]1, 2]$. By surpassing the point 2 by the center weight of object B and till the point 3 or more, the qualitative distance is described as far. The center weight of object 1 still at point 0 and the center weight of the object B vary in the interval $]2, 3]$.

3.4 A composite spatial relation between two objects

A composite spatial relation between two objects A and B is a relation composed of multiple types of spatial relations Fig. 7; it is used to enhance the information about a relative spatial description between a reference object A and another object B. In this abstract spatial description, a combination of the three previously mentioned spatial information is used

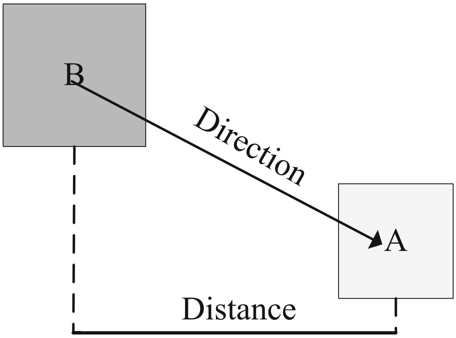


Fig. 7 A composite spatial relation

to provide a full description of a spatial relation. Given two objects A and B, the composite spatial relation between these two objects is defined as follows:

R={Top, Dir, Dis} where:

- Top represents the qualitative topology according to RCC8 of the given two objects
- Dir represents the qualitative direction of the center-weights of the two objects
- Dis represents the qualitative distance between the center-weights of the two objects

The next figure shows a sample representation of a spatial relation between two disjoint (according to RCC8) objects A and B.

3.5 An XML to PROLOG representation

In order to process spatial information available in an XMS document by our inference system, a transition from XML elements to PROLOG predicates is needed. The transition can be done while reading the XMS source file so the XML specification is translated into predicates to be processed by PROLOG later. In the next element, we provide how composite spatial relations are defined in our XMS document and how are they presented in our inference system.

In XMS, the composite spatial relations are represented according to the next portion of the XSD schema:

Where:

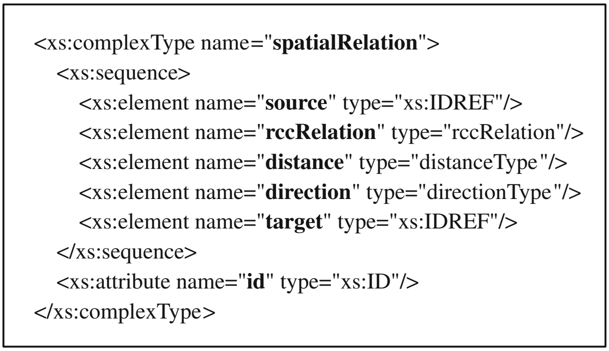
- source: defines the id of the source object.
- rccRelation: defines the topology of the two objects
- distance: defines the qualitative distance between the center-weights of the two objects
- direction: defines the qualitative direction of the center weight of the target object according to the center weight of the source object.
- target: defines the id the target object.

According to the previous schema the relation between object A and object B represented in Fig. 8 is defined in XMS as follows:

Given the id=o1 for the object A and the id=o2 for the object B, the XML portion that describe the composite spatial relation is shown in Fig. 9.

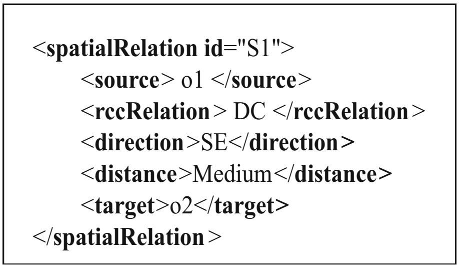
A composite spatial relation that is present in XMS like we mentioned previously is translated to a PROLOG predicate according to the form presented in Fig. 10.

The previous example of the XML composite spatial relation is translated according to the previous figure into a PROLOG predicate show in Fig. 11.



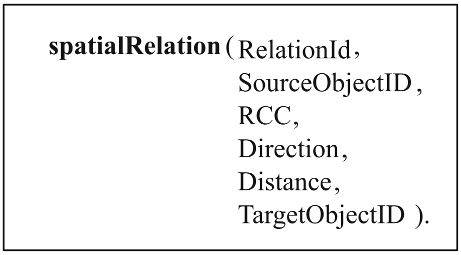
```
<xs:complexType name="spatialRelation">
   <xs:sequence>
      <xs:element name="source" type="xs:IDREF"/>
      <xs:element name="rccRelation" type="rccRelation"/>
      <xs:element name="distance" type="distanceType"/>
      <xs:element name="direction" type="directionType"/>
      <xs:element name="target" type="xs:IDREF"/>
   </xs:sequence>
   <xs:attribute name="id" type="xs:ID"/>
</xs:complexType>
```

Fig. 8 A composite spatial relation XMS schema



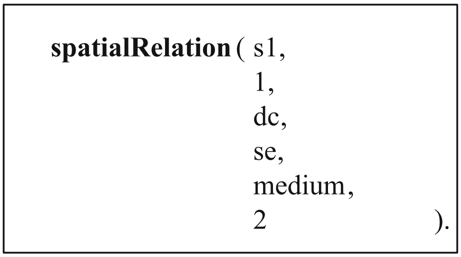
```
<spatialRelation id="S1">
      <source> o1 </source>
      <rccRelation> DC </rccRelation>
      <direction>SE</direction>
      <distance>Medium</distance>
      <target>o2</target>
</spatialRelation>
```

Fig. 9 A sample composite spatial relation in XMS



```
spatialRelation ( RelationId,
                  SourceObjectID ,
                  RCC,
                  Direction,
                  Distance,
                  TargetObjectID ).
```

Fig. 10 The form of a composite spatial relation in PROLOG



```
spatialRelation ( s1,
                  1,
                  dc,
                  se,
                  medium,
                  2              ).
```

Fig. 11 A sample of a composite spatial relation in PROLOG

4 Spatial specification manipulation

A specification should be manipulated and adapted in order to make it suitable to a certain situation or profile; manipulations vary on several levels and according to several situations. The most used manipulation is to reduce either the spatial dimensions or temporal dimensions of a multimedia document so it fits and can be presented in a target device or situation. In the proposed approach, both reduction and stretching of a specification is considered in the inference system. Here we show how to reduce a spatial specification or stretch it to fit a certain profile.

According to the proposed spatial specification model, three main steps are used to either reduce or stretch the specification.

4.1 Distances refining

The first step in manipulating is the distance refining by either augmenting or reducing the distance depending on the situation. To do so, two ways browsing of the distances line allows us to achieve the needed transformation as shown in Fig. 12.

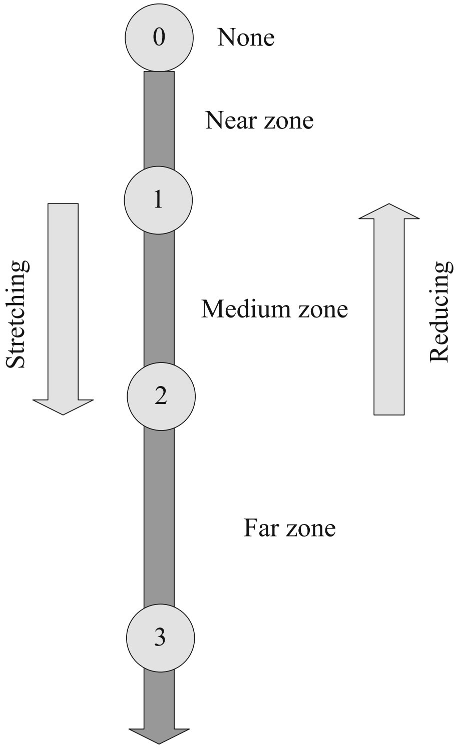


Fig. 12 Distances manipulations

Table 2 Distances manipulation predicates

Type	Predicate
Stretching	refineDistance(stretch,none,near).
Stretching	refineDistance(stretch,near,medium).
Stretching	refineDistance(stretch,medium,far).
Stop	refineDistance(stretch,far,far).
Reducing	refineDistance(reduce,far,medium).
Reducing	refineDistance(reduce,medium,near).
Reducing	refineDistance(reduce,near,none).
Stop	refineDistance(reduce,none,none).

According to Fig. 12, we can count four main manipulations of distances in each way. These manipulations are represented in PROLOG predicates in Table 2. Manipulations of type 'stop' define the end of either reducing or stretching a distance of a spatial relation thus, when we need to manipulate a distance we may add to it in order to either enlarge or reduce the size of a composition. Two possible cases arise when we face a composition where the profile defines a device that is either too large for displaying the composition or too small.

Stretching distances where permissible (In some other cases the author may choose to restrict certain distances from being stretched or reduced) helps us to achieve a consistent composition where the final display is too large for the original composition. On the other hand, reducing distances helps us to achieve a consistent composition where the final display is too small for the original composition.

In the case where after all the initial distances are reduced to a minimum 'none' or stretched to a maximum 'far', if the resulted specification does not match the profile then the adaptation of the spatial specification takes the next step.

Algorithm 1 is a simplified form of the spatial distances refining algorithm.It is based on the next basic steps:

- It starts by verifying the first spatial relation in the specification where the distance can be refined. This can be done by verifying if the result of refining the distance in the spatial relation is the same and no further change can be done. The first most spatial relation that can be refined is then taken as r.
- r is then refined into a new spatial relation r'.
- Another part of the algorithm verifies if there are any affected distances in the specification and create a new spatial specification s', it contains all non-refined relations and the refined ones, also all affected relations.
- Check if the new specification s' does in fact match the profile and delivers the specification and exit.
- In case the new specification does not match the profile, the algorithm then checks if all the relations in s' are refined to maximum and exit.
- If at least on spatial relation in s' can be refined the specification s' is then refined to a new one.

Algorithm 1 Distance refining

```
Input: profile $p$, document $d$, spatial specification $s$, adaptation direction $dir$
Output: adapted specification $s'$
$i \leftarrow 0$
$n \leftarrow$ number of relations in $s$
while refineDistance($dir$, $s[i](distance)$, $s[i](distance)$) & $i < n$ do
    $i \leftarrow i++$
end while
$distance = s[i](distance)$
refineDistance($dir$, $distance$, $refinedDistance$)
$r' \leftarrow s[i]$
$r'(distance) \leftarrow refinedDistance$
generate the affected distances
$s' \leftarrow s$
$s'[i] = r'$
update all affected distances in $s'$
if match($p$, $s$) then
    exit
else
    $j \leftarrow 0$
    $r \leftarrow s'[j]$
    while refineDistance($dir$, $s'[i](distance)$, $s'[i](distance)$) & $j < n$ do
        $j \leftarrow j++$
    end while
    if $j = n$ all distances are refined to maximum then
        exit
    else
        $s' \leftarrow$ refineDistances($p$, $s'$, $dir$)
    end if
end if
```

4.2 Directions rotation

If after refining all the distances to a maximum, the spatial specification still does not match the profile. The next step of the manipulation consists of rotating the different directions; either the width or the length of the specification is affected. The rotation of a directional relation is done according to either a clockwise direction or a counter-clockwise direction as shown in the Fig. 13.

Different possibilities when it comes to spatial directions, the final possible rotation of a given relation depends on the first state of the direction. We count from north to north a full rotation meanwhile; the adaptation requires only basic steps of rotation. According to the previous figure we count nine different basic possible rotation steps in each way (clockwise, or counter-clockwise) including the return step (the step that takes us back to the initial direction) and the non-rotation step where the initial objects have the same center weight.

The basic rotation steps are represented PROLOG predicates in Tables 3 and 4 starting from the east direction. Table 3 for clockwise rotation and Table 4 for clockwise rotation.The type stop defines the end of either clockwise or counter-clockwise rotation of the direction of a spatial relation.

Algorithm 2 Directions rotation

```
Input: profile $p$, document $d$, spatial specification $s$, adaptation direction $dir$
Output: adapted specification $s'$
$i \leftarrow 0$
$n \leftarrow$ number of relations in $s$
while rotateDirection($dir$, $s[i](direction)$, $s'[i](direction)$) & $i < n$ do
    $i \leftarrow i++$
end while
$r \leftarrow s[i]$
$direction \leftarrow r(direction)$
rotateDirection($dir$, $direction$, $rotatedDirection$)
$r' \leftarrow r$
$r'(direction) \leftarrow rotatedDirection$
$s' \leftarrow s$
$s'[i] \leftarrow r'$
update all affected directions in $s'$
if match($p$, $s$) then
    exit
else
    $j \leftarrow 0$
    while rotateDirection($dir$, $s'[j](direction)$, $s'[j](direction)$) & $j < n$ do
        $j \leftarrow j++$
    end while
    if $j = n$ all direction are rotated to maximum then
        exit
    else
        $s' \leftarrow$ rotateDirections($p$, $s'$, $dir$)
    end if
end if
```

Algorithm 2 is just like Algorithm 1 a simplified form of the spatial direction rotation algorithm.It is based on the next basic steps:

- It starts by verifying the first spatial relation in the specification where the direction can be rotated. this can be done by verifying if the result of rotation of the distance in

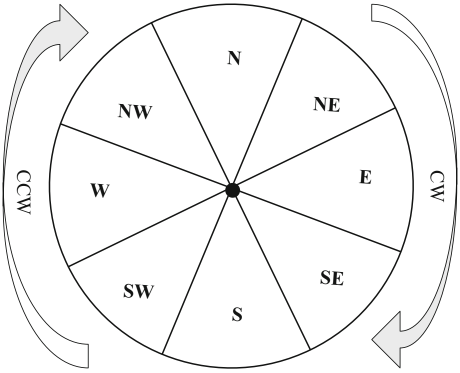


Fig. 13 directional relations manipulations

Table 3 Clockwise directions manipulation predicates

Type	Predicate
Clockwise	rotateDirection(cw,e,se).
Clockwise	rotateDirection(cw,se,s).
Clockwise	rotateDirection(cw,s,sw).
Clockwise	rotateDirection(cw,sw,w).
Clockwise	rotateDirection(cw,w,nw).
Clockwise	rotateDirection(cw,nw,n).
Clockwise	rotateDirection(cw,n,ne).
Clockwise	rotateDirection(cw,ne,e).
Stop	rotateDirection(cw,none,none).

a spatial relation is the same and no further change can be done. The cases where no further change can be done are the predicate having the type "STOP" in Tables 3 and 4. The first most spatial relation that can be rotated is then taken as r.

- r is then rotated into a new spatial relation r'.
- Another part of the algorithm verifies if there are any affected directions in the specification and create a new spatial specification s', it contains all non-rotated relations and the rotated ones, also all affected relations.
- Check if the new specification s' does in fact match the profile, deliver the specification and exit.
- In case the new specification does not match the profile, the algorithm then checks if all the relations in s' are rotated to maximum and exit.
- If at least on spatial relation in s' can be rotated the specification s' is then refined to a new one.

4.3 Topological refining

In a final try to adapt the spatial specification and if all the initial directions are rotated to an end direction and the resulted specification does not match the profile the inference system takes the specification to a final step. It consists of adapting the topology of the spatial specification according a neighborhood graph [12]

Table 4 Counter-clockwise directions manipulation predicates

Type	Predicate
stop	rotateDirection(ccw,none,none).
Counter-clockwise	rotateDirection(ccw,e,ne).
Counter-clockwise	rotateDirection(ccw,ne,n).
Counter-clockwise	rotateDirection(ccw,n,nw).
Counter-clockwise	rotateDirection(ccw,nw,w).
Counter-clockwise	rotateDirection(ccw,w,sw).
Counter-clockwise	rotateDirection(ccw,sw,s).
Counter-clockwise	rotateDirection(ccw,s,se).
Counter-clockwise	rotateDirection(ccw,se,e).

Table 5 Topological reducing predicates

Type	Predicate
Reduce	refineTopology(dc,ec,_,reduce).
Reduce	refineTopology(ec,po,_,reduce).
Reduce	refineTopology(po,tpp,S,reduce):- spatialRelation(S,O1,po,_,_,O2), spatialProperty(_,O1,[MinH1,MinW1],_), spatialProperty(_,O2,[MinH2,MinW2],_), MinH1<MinH2,MinW1<MinW2.
Reduce	refineTopology(po,tppi,S,reduce):- spatialRelation(S,O1,po,_,_,O2), spatialProperty(_,O1,[MinH1,MinW1],_), spatialProperty(_,O2,[MinH2,MinW2],_), MinH1>MinH2,MinW1>MinW2.
Reduce	**refineTopology**(po,eq,S,reduce):- spatialRelation(S,O1,po,_,_,O2), spatialProperty(_,O1,[MinH1,MinW1],_), spatialProperty(_,O2,[MinH2,MinW2],_), MinH1=MinH2,MinW1=MinW2.

Just like the distances line, the direction of browsing the neighborhood graph defines either stretching or reducing the space occupied by the two objects. The PROLOG representation of the RCC8 neighborhood graph requires complex clauses in order to define the conditions for the transition between each two nodes; it depends on the size of the source object compared to the target object in the following cases:

- Reducing from DC to EC and from EC to PO
- Reducing from PO to either: TPP, EQ or TPPi depending on the two objects of a spatial relation
- Stretching from NTPP to TPP
- Stretching from EQ to PO
- Stretching from NTPPi to TPPi
- Stretching from either TPP or TPPi to PO
- Stretching from PO to EC and from EC to DC

Table 5 shows predicates for topological reduction of a spatial relation, there are five different possible reductions explained as follows:

- From DC to EC without further conditions
- From EC to PO without further conditions
- From PO to TPP in the case of $O1$ occupies lesser space than $O2$
- From PO to TPPi in the case of $O2$ occupies lesser space than $O1$
- From PO to EQ in the case of $O1$ occupies the same space as $O2$

Table 6 shows the predicates where further reducing of the space occupied by two objects using topology refining, the following cases are the equivalent of the predicates:

- The topology is PO (partially overlapped) and neither object A nor object B is bigger than the other.
- The topology is either: TPP, TPPi, EQ, NTPP, NTPPi.

Table 6 Topological stopping for when reducing predicates

Type	Predicate
Stop	refineTopology(po,po,S,reduce):- spatialRelation(S,O1,po,_,_,O2), spatialProperty(_,O1,[MinH1,MinW1],_), spatialProperty(_,O2,[MinH2,MinW2],_), ((MinH1<MinH2,MinW1>MinW2); (MinH1>MinH2,MinW1<MinW2)).
Stop	refineTopology(tpp,tpp,_,reduce).
Stop	refineTopology(tppi,tppi,_,reduce).
Stop	refineTopology(eq,eq,_,reduce).
Stop	refineTopology(ntpp,ntpp,_,reduce).
Stop	refineTopology(ntppi,ntppi,_,reduce).

Stretching according to the neighborhoud graph (Fig. 14) is simpler than reducing. Table 7 shows the following stretching predicates:

- From NTPP to TPP
- From NTPPi to TPPi
- From EQ to PO
- From TPP to PO
- From TPPi to PO
- From PO to EC
- From EC to DC

Table 7 also shows where no further stretching can be performed according to topology refining, when stretching the DC topology (disjoint) can no longer be stretched by changing the topology.

The Algorithm 3 is a simplified form of the spatial topology refining algorithm.

Algorithm 3 is just like Algorithms 1 and 2 a simplified form of the spatial topology refining algorithm.It is based on the next basic steps:

- It starts by verifying the first spatial relation in the specification where the topology can be refined. This can be done by verifying if the result of refining of the topology in a spatial relation is a stop and no further reduction or stretching can be done. The cases where no further change can be done are the predicate having the type “STOP” in

Table 7 Topological stretching predicates

Type	Predicate
Stretch	refineTopology(ntpp,tpp,_,stretch).
Stretch	refineTopology(ntppi,tppi,_,stretch).
Stretch	refineTopology(eq,po,_,stretch).
Stretch	refineTopology(tpp,po,_,stretch).
Stretch	refineTopology(tppi,po,_,stretch).
Stretch	refineTopology(po,ec,_,stretch).
Stretch	refineTopology(ec,dc,_,stretch).
Stop	refineTopology(dc,dc,_,stretch).

Tables 6 and 7. The first most spatial relation that can be topologically refined is then taken as r.

- r is then topologically refined into a new spatial relation r'.
- Another part of the algorithm verifies if there are any affected topologies in the specification and create a new spatial specification s' which contains all non-refined relations and the refined ones, also all affected relations.
- Check if the new specification s' does in fact match the profile, deliver the specification and exit.
- In case the new specification does not match the profile, the algorithm then checks if all the relations in s' are refined to maximum and exit.
- If at least on spatial relation in s' can be further refined the specification s' is then refined to a new one.

Algorithm 3 Topology refining

```
Input: profile $p$, document $d$, spatial specification $s$, adaptation direction $dir$
Output: adapted specification $s'$
$i \leftarrow 0$
$n \leftarrow$ number of relations in $s$
while refineTopology($s[i](topology), s[i](topology), dir$)) & $i < n$ do
    $i \leftarrow i++$
end while
$topology = s[i](topology)$
refineTopology($dir, topology, refinedTopology$)
$r' \leftarrow s[i]$
$r'(topology) \leftarrow refinedTopology$
generate the affected topology
$s' \leftarrow s$
$s'[i] = r'$
update all affected topologies in $s'$
if match($p, s$) then
    exit
else
    $j \leftarrow 0$
    $r \leftarrow s'[j]$
    while refineTopology($s'[i](topology), s'[i](topology),$) & $j < n$ do
        $j \leftarrow j++$
    end while
    if $j = n$ all topologies are refined to maximum then
        exit
    else
        $s' \leftarrow$ refineTopologies($p, s', dir$)
    end if
end if
```

The Fig. 14 represents a modified RCC8 neighborhood graph used to refine the topology of a spatial specification. The dashed ends of the edges in the Fig. 14 are modified in the original neighborhoud graph [12], they mean we can not go in that direction. The cases

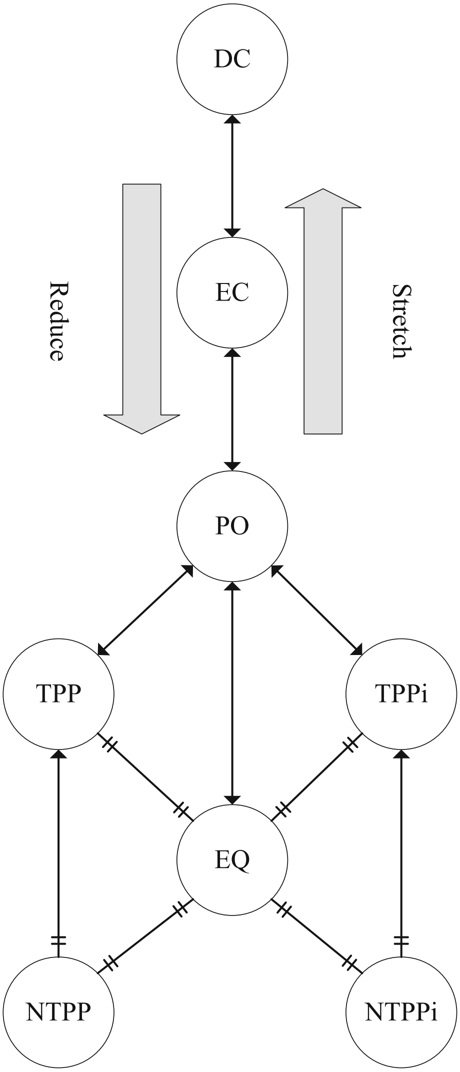


Fig. 14 Modified RCC8 neighborhoud graph

where topological adaptation can do nothing are explained previously in the 'STOP' types of predicates.

5 Spatial adaptation algorithms

Previously, some basic manipulations over a spatial specification is discussed in Algorithms 1, 2 and 3, the main Algorithm 4 is an assembly of the previous algorithms.

Algorithm 4 Main adaptation algorithm

```
Input: profile $p$, document $d$, spatial specification $s$, adaptation direction $dir$
Output: adapted specification $s'$
if match($p, s$) then
    exit
else
    $s' \leftarrow$ refineDistances($p, s, dir$)
    if match($p, s'$) then
        exit
    else
        $s' \leftarrow$ rotateDirections($p, s, dir, 0$)
        if match($p, s'$) then
            exit
        else
            $s' \leftarrow$ refineTopologies($p, s', dir$)
            if match($p, s'$) then
                exit
            end if
        end if
    end if
end if
delivers the most refined version or take further actions and exit
```

The algorithm passes the initial spatial specification on a three-stage refining on the spatial dimension, these stages are based on Algorithms 1, 2 and 3 for spatial manipulation. Algorithm 4 is a simplified version of the adaptation algorithm. The algorithm consists of these main stages:

- The first stage consist of verifying if the spatial specification is already conform to the profile and exits the algorithm in that case. Otherwise, the algorithm passes to the second stage.
- In the second stage the algorithm refines the distances in the spatial specification in order either to reduce or stretch the space occupied by the final percept.

Definition 6 A percept according to the Oxford dictionary is a mental concept that is developed as a consequence of the process of perception. In our abstract document, a percept is the final composition of a set of media elements in space and time; this composition is the one that the final user is going to perceive. The original document specification is the target percept, which the author intends originally to deliver to the final user. After applying the adaptation algorithm, the result might or might not achieve a consistent composition in space and time that fits the constraints of the final user. This leads to create something in between that neither give the specification as defined by the author nor respect all the constraints of the user and delivers a consistent composition that is called a percept.

- In case after refining distances the resulted specification s' matches the profile, s' is simply delivered. Otherwise the algorithm takes thee adaptation to the next level.
- If refining the distances would not produce a fairly matched specification, the specification s' resulted from previous step is passed through the directions rotation.

```
profile(profileName).
device(profileName,screenID,memoryID,batteryID).
screen(screenID,physicalSizeInches,widthPixel,
        heightPixel,pixelDensity).
battery(batteryID,batteryPercentage).
memory(memoryID,availableMemory).
```

Fig. 15 A simple profile PROLOG predicates

- In case after rotating the directions the resulted specification s' matches the profile, s' is then delivered. Otherwise the algorithms passes to the final try to adapt s'.
- The final stage consists of adapting the topology of the spatial specification s' resulted in the previous stage.
- If the resulted specification of topology adaptation s' does not match the profile, either the final spatial specification s' is delivered or according to the final user consent from the profile the original specification is delivered instead.

Advanced mechanisms might be used after all the tries of adapting of the spatial specification. Some mechanisms to refine a specification further including and not limited to: temporal decomposition, hypermedia decomposition as if the specification is split into two or more slides where every slide matches the spatial requirements [31].

5.1 A simple profile for adaptation

In order to adapt a multimedia document a reasoning about the document and the situation in which the document will be used is required. To do so, a profile must be used to get information about the device, location and some user preferences [10, 21, 33, 35]. In this work, we used a simple profile that is translated from XML to PROLOG. The profile to be used in the inference system defines the screen, battery, and memory information of the target device so the system can decide what transformations and manipulations can be done on the specification in order to match the profile. Further information are required in a real life scenario, several works have tried to define a profile like the well known CC/PP standard [33], others based on this standard or tried to suppress its limitations

The next set of predicates defines a simple profile (Fig. 15):

A sample smart-phone profile should look like (Fig. 16):

Fig. 16 A simple smart-phone profile PROLOG predicates

```
profile(smartPhone).
device(smartPhone,s1,m1,b1).
screen(s1,4.8,360,640,2).
battery(b1,20).
memory(m1,60).
```

6 Preliminary results and discussion

In order to make the goal of developing further XMS and the algorithms present in this paper clear, a simple example is used in this section to showcase the manipulation algorithms one by one.

In Fig. 17 a sample abstract layout of a web page is shown, this layout can be extended to a different layout with more details and therefore giving its specification a higher granularity (see Definition 5). To keep the example simple for understanding we chose only to keep the outer layout, the following elements presents examples of applying the algorithms on the layout.

6.1 The XMS specification

In our approach the author starts by writing a specification of the original document using XMS, a sample specification corresponding to the simple page layout shown in Fig. 17 is composed of three main main elements:

- The XMS header containing general information about the document and possible alternatives.
- The objects declaration containing references and details about objects in the original document.
- The specification part containing both spatial properties and relations.

The header of the XMS specification of the simple page layout is shown in Fig. 18. The header in this case contains only the target original document.

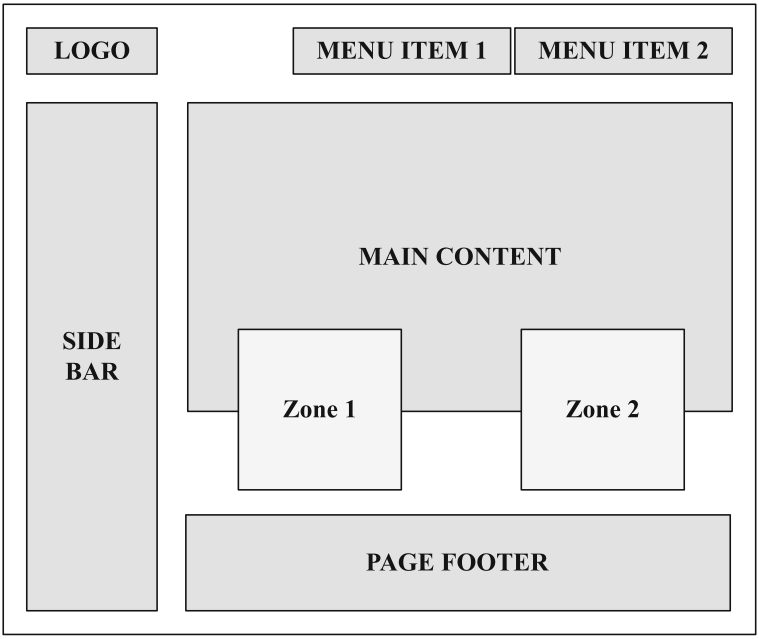
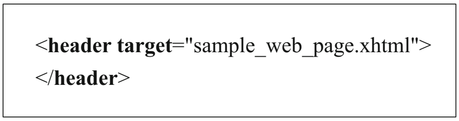


Fig. 17 A simple web page layout

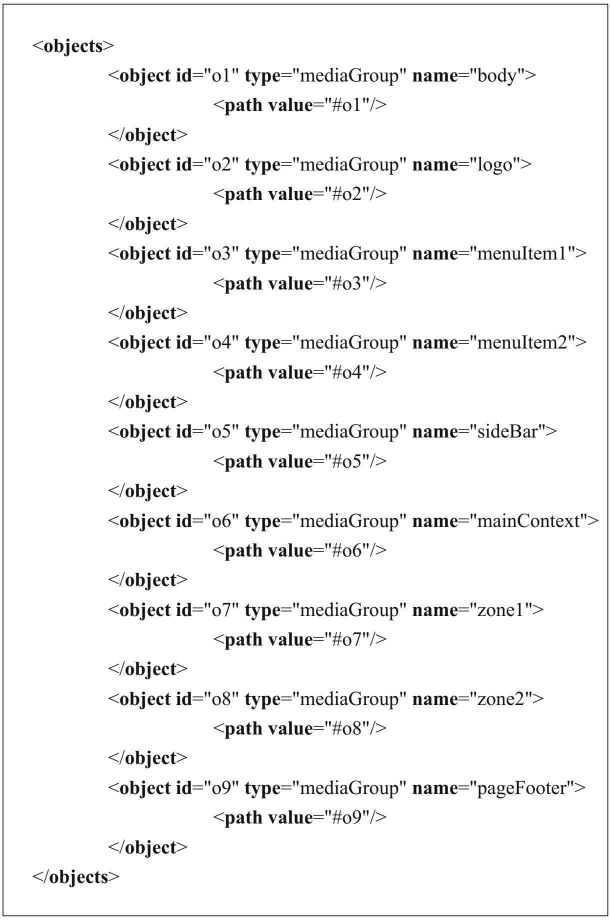


```
<header target="sample_web_page.xhtml">
</header>
```

Fig. 18 XMS specification header

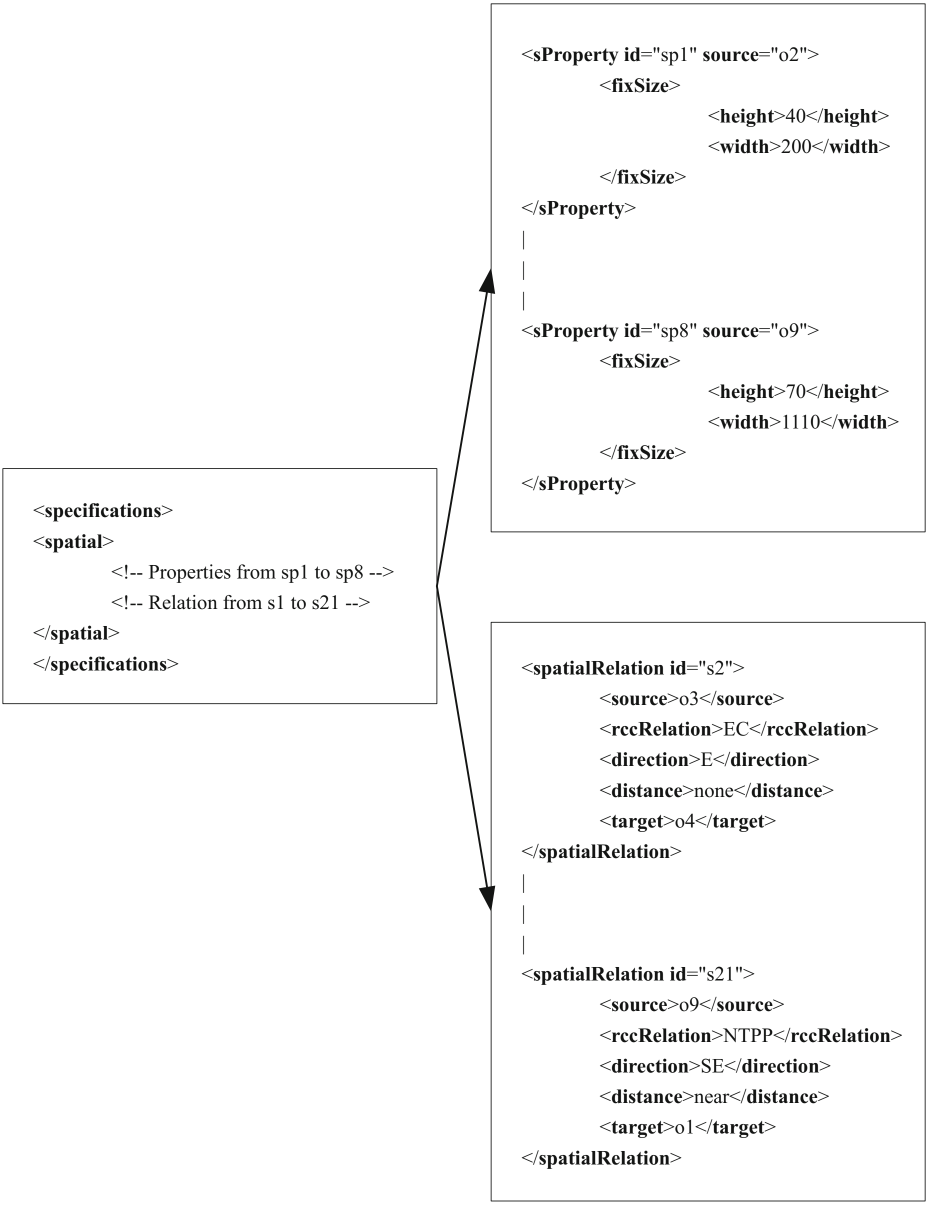
The object declaration part of the specification is shown in Fig. 19. The object declaration references objects in the original document, in this case using HTML identifiers #*id*. The type attribute defines whether the object being defined is a simple object or a media group. A media group type means the object is a complex object that may contain other components such as multiple HTML elements embedded in a DIV element.

A sample of the spatial specification is show in Fig. 20 both the spatial properties and the spatial relations are shown, eight spatial properties are used to describe the objects of



```
<objects>
        <object id="o1" type="mediaGroup" name="body">
                <path value="#o1"/>
        </object>
        <object id="o2" type="mediaGroup" name="logo">
                <path value="#o2"/>
        </object>
        <object id="o3" type="mediaGroup" name="menuItem1">
                <path value="#o3"/>
        </object>
        <object id="o4" type="mediaGroup" name="menuItem2">
                <path value="#o4"/>
        </object>
        <object id="o5" type="mediaGroup" name="sideBar">
                <path value="#o5"/>
        </object>
        <object id="o6" type="mediaGroup" name="mainContext">
                <path value="#o6"/>
        </object>
        <object id="o7" type="mediaGroup" name="zone1">
                <path value="#o7"/>
        </object>
        <object id="o8" type="mediaGroup" name="zone2">
                <path value="#o8"/>
        </object>
        <object id="o9" type="mediaGroup" name="pageFooter">
                <path value="#o9"/>
        </object>
</objects>
```

Fig. 19 XMS specification object declaration



```
<specifications>
<spatial>
        <!-- Properties from sp1 to sp8 -->
        <!-- Relation from s1 to s21 -->
</spatial>
</specifications>
```

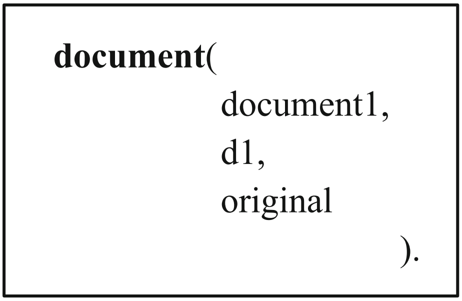
```
<sProperty id="sp1" source="o2">
        <fixSize>
                    <height>40</height>
                    <width>200</width>
        </fixSize>
</sProperty>
|
|
|
<sProperty id="sp8" source="o9">
        <fixSize>
                    <height>70</height>
                    <width>1110</width>
        </fixSize>
</sProperty>
```

```
<spatialRelation id="s2">
        <source>o3</source>
        <rccRelation>EC</rccRelation>
        <direction>E</direction>
        <distance>none</distance>
        <target>o4</target>
</spatialRelation>
|
|
|
<spatialRelation id="s21">
        <source>o9</source>
        <rccRelation>NTPP</rccRelation>
        <direction>SE</direction>
        <distance>near</distance>
        <target>o1</target>
</spatialRelation>
```

Fig. 20 XMS spatial specification sample

the specification other than the reference object $o1$ which is the main body of the sample web page. The body is as flexible as the whole composition, it is used to better describe positions of different elements.

For the spatial relations, we have found that twenty one spatial relation can describe fairly the sample web page. Not all properties and relations are shown in the figure for ease of understanding.



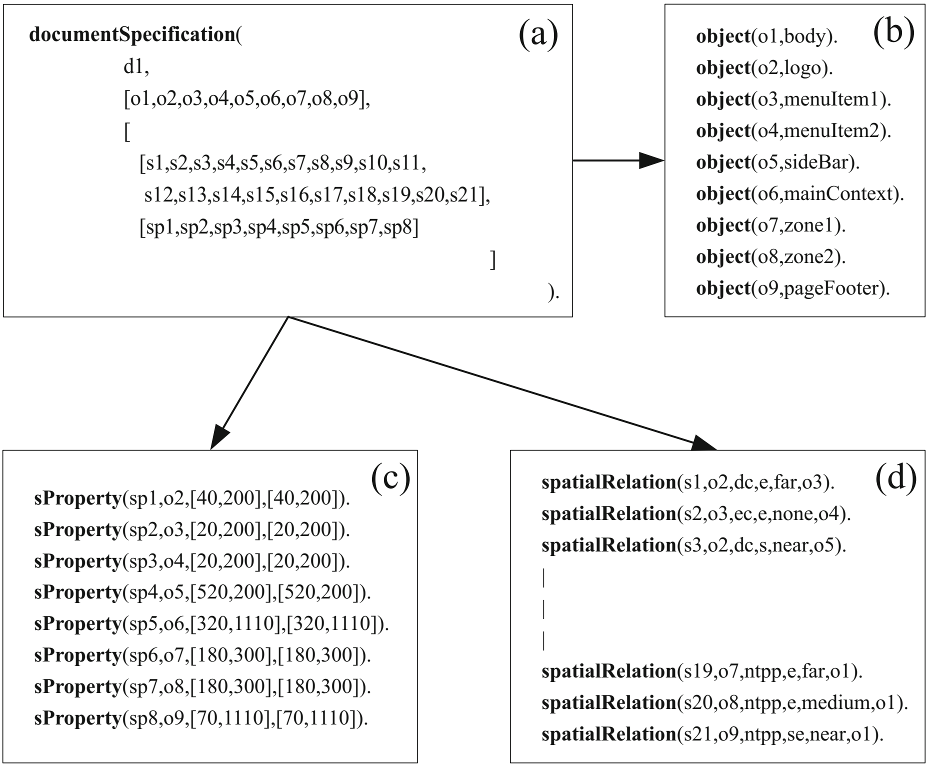
```
document(
            document1,
            d1,
            original
                        ).
```

Fig. 21 XMS main PROLOG predicate

6.2 Prolog representation

In order to process the XMS specification, a minimal version of the specification is used as a set of PROLOG predicates (facts). Just like XMS specification, same main elements exists in PROLOG predicates as follows:

- The XMS main predicate containing general information about the document, possible alternatives are defined in other predicates.
- The objects declaration containing references and other details are represented in a set of predicates.



```
documentSpecification(
            d1,
            [o1,o2,o3,o4,o5,o6,o7,o8,o9],
            [
              [s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,
               s12,s13,s14,s15,s16,s17,s18,s19,s20,s21],
              [sp1,sp2,sp3,sp4,sp5,sp6,sp7,sp8]
                                                    ]
                                                      ).
```

```
object(o1,body).
object(o2,logo).
object(o3,menuItem1).
object(o4,menuItem2).
object(o5,sideBar).
object(o6,mainContext).
object(o7,zone1).
object(o8,zone2).
object(o9,pageFooter).
```

```
sProperty(sp1,o2,[40,200],[40,200]).
sProperty(sp2,o3,[20,200],[20,200]).
sProperty(sp3,o4,[20,200],[20,200]).
sProperty(sp4,o5,[520,200],[520,200]).
sProperty(sp5,o6,[320,1110],[320,1110]).
sProperty(sp6,o7,[180,300],[180,300]).
sProperty(sp7,o8,[180,300],[180,300]).
sProperty(sp8,o9,[70,1110],[70,1110]).
```

```
spatialRelation(s1,o2,dc,e,far,o3).
spatialRelation(s2,o3,ec,e,none,o4).
spatialRelation(s3,o2,dc,s,near,o5).
|
|
|
spatialRelation(s19,o7,ntpp,e,far,o1).
spatialRelation(s20,o8,ntpp,e,medium,o1).
spatialRelation(s21,o9,ntpp,se,near,o1).
```

Fig. 22 XMS PROLOG specification

Fig. 23 Original spatial relations before any manipulations

```
spatialRelation(s1, o2, dc, e, far, o3).
spatialRelation(s2, o3, ec, e, none, o4).
spatialRelation(s3, o2, dc, s, near, o5).
spatialRelation(s4, o3, dc, s, near, o6).
spatialRelation(s5, o4, dc, s, near, o6).
spatialRelation(s6, o5, dc, e, near, o6).
spatialRelation(s7, o6, po, s, near, o7).
spatialRelation(s8, o6, po, s, near, o8).
spatialRelation(s9, o7, dc, e, near, o8).
spatialRelation(s10, o5, dc, e, medium, o7).
spatialRelation(s11, o5, dc, e, near, o9).
spatialRelation(s12, o7, dc, s, near, o9).
spatialRelation(s13, o8, dc, s, near, o9).
spatialRelation(s14, o2, ntpp, nw, near, o1).
spatialRelation(s15, o3, ntpp, n, near, o1).
spatialRelation(s16, o4, ntpp, ne, near, o1).
spatialRelation(s17, o5, ntpp, sw, near, o1).
spatialRelation(s18, o6, ntpp, e, near, o1).
spatialRelation(s19, o7, ntpp, e, far, o1).
spatialRelation(s20, o8, ntpp, e, medium, o1).
spatialRelation(s21, o9, ntpp, se, near, o1).
```

- The specification part containing both spatial properties and relations is represented using a set of predicates.

The main predicate describing the document is shown in Fig. 21, the third term ‘original’ is used to describe the specification version. In case alternatives are given to the entire document, such as a different specification for a different profile is given this term is named accordingly.

The entire XMS specification in PROLOG predicates is shown in Fig. 22, except the spatial relations were simplified for ease of presentation in part (d) of the figure. Part (a) shows the main predicate where the entire specification is defined, part (b) shows the definition of the set of predicates which defines objects of XMS specification. Part (c) shows the spatial properties according to the XMS specification for each of the objects, except the reference object which is the ‘body’ object.

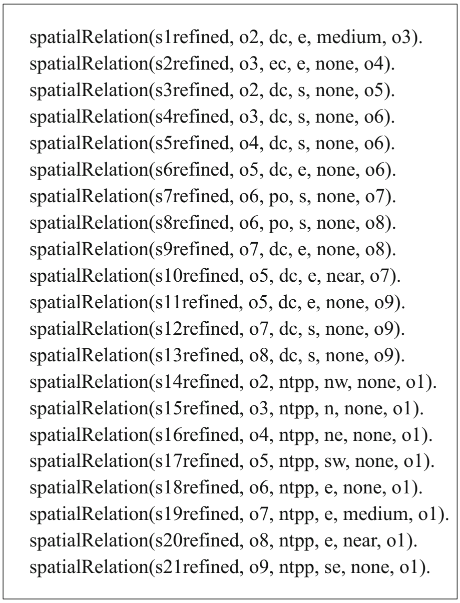
6.3 Sample results

In this part we show some sample results using a unique iteration of each manipulation Algorithms 1, 2 and 3 presented in this paper. Figure 23 shows the original relations before any manipulations.

```
?- time(perceptDistancesRefine(reduce,d1,smartPhone,X)).
% 220 inferences, 0.000 CPU in 0.002 seconds (?% CPU, Infinite Lips)
X = d1refined.
```

Fig. 24 Distance refining predicate call in PROLOG

Fig. 25 Spatial relations after distance refining call

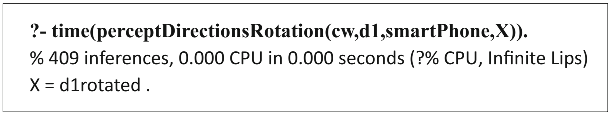


```
spatialRelation(s1refined, o2, dc, e, medium, o3).
spatialRelation(s2refined, o3, ec, e, none, o4).
spatialRelation(s3refined, o2, dc, s, none, o5).
spatialRelation(s4refined, o3, dc, s, none, o6).
spatialRelation(s5refined, o4, dc, s, none, o6).
spatialRelation(s6refined, o5, dc, e, none, o6).
spatialRelation(s7refined, o6, po, s, none, o7).
spatialRelation(s8refined, o6, po, s, none, o8).
spatialRelation(s9refined, o7, dc, e, none, o8).
spatialRelation(s10refined, o5, dc, e, near, o7).
spatialRelation(s11refined, o5, dc, e, none, o9).
spatialRelation(s12refined, o7, dc, s, none, o9).
spatialRelation(s13refined, o8, dc, s, none, o9).
spatialRelation(s14refined, o2, ntpp, nw, none, o1).
spatialRelation(s15refined, o3, ntpp, n, none, o1).
spatialRelation(s16refined, o4, ntpp, ne, none, o1).
spatialRelation(s17refined, o5, ntpp, sw, none, o1).
spatialRelation(s18refined, o6, ntpp, e, none, o1).
spatialRelation(s19refined, o7, ntpp, e, medium, o1).
spatialRelation(s20refined, o8, ntpp, e, near, o1).
spatialRelation(s21refined, o9, ntpp, se, none, o1).
```

Figure 24 shows a sample query of the distance refining Algorithm 1. The resulted new document specification $d1refined$ contains the set of new refined relations, we concatenated the original document ID with the term ‘refined’ to differentiate the original and the refined specification. Figure 25 shows the set of new relations in the refined specification, the profile ‘smartPhone’ used in the query is presented in the knowledge base and can refer to the sample profile presented in Fig. 16.

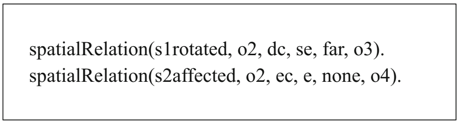
Figure 26 shows a sample query of the direction rotation Algorithm 2. The resulted new document specification “$d1rotated$” contains the set of new rotated and affected relations, we concatenated the original document ID with the term ‘rotated’ to differentiate the original and the rotated specification. Figure 27 shows the set of new relations in the new specification. Not all relations are shown, only the rotated and the affected relations are asserted in the system for better optimisation. In the query of the rotation algorithm predicate we used the original document specification, usually the rotation predicate is used after calling the distance refining algorithm predicate.

Figure 28 shows a sample query of the topology refining Algorithm 2. The resulted new document specification $d1toporefine$ contains the set of new refined relations, we concatenated the original document ID with the term ‘toporefine’ to differentiate the original and the refined specification. Figure 29 shows the set of new relations in the new specification. Not all relations are shown, only the refined relations are asserted in the system for better optimisation. Before calling the topology refining algorithm predicate we used the



```
?- time(perceptDirectionsRotation(cw,d1,smartPhone,X)).
% 409 inferences, 0.000 CPU in 0.000 seconds (?% CPU, Infinite Lips)
X = d1rotated .
```

Fig. 26 Direction rotation predicate call in PROLOG



```
spatialRelation(s1rotated, o2, dc, se, far, o3).
spatialRelation(s2affected, o2, ec, e, none, o4).
```

Fig. 27 Spatial relations rotated or affected after direction rotation call

original document specification, usually the topology refining predicate is used after calling the direction rotation predicate.

6.4 XMS in comparison with existing solutions

The most closely similar work is that of Euzenat et al. [12] the authors take a similar approach by designing a new intermediate document called MADEUS. They use this document as a canal between a high-level XML-based multimedia document (source document) and a low-level rendering format. Unlike XMS, MADEUS does not only describe the original document. Instead, it is a whole transition that contains both content and specification. The main inconvenience to this is the level of separation. By using MADEUS, there will be no separation between the content and the adaptation. Meanwhile, XMS takes the original document out of the equation. Also, the fact that MADEUS is not based on the higher level standardized multimedia XML-based document do not allow it to go further and become a generalized solution for multimedia adaptation in general.

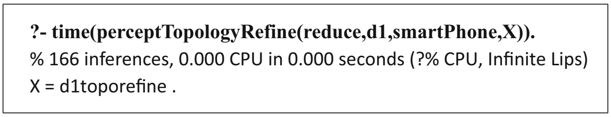
Another closely similar work is that of He et al. [20] where authors proposed a flexible content adaptation system using a rule-based approach called XADAPTOR. They developed novel adaptation techniques for structure object HTML table, the proposed adaptation system is based on the reorganization of table cells to adapt a content of an HTML document. This technique is more focused on only a specific part of HTML instead of a generalized solution for content adaptation. The authors claim the solution is extensible by adding rules in the PROLOG based inference system. However unlike XMS, the separation between the content and adaptation is not addressed, thus limiting the applications of the techniques proposed by the authors.

The next Table 8 provide a comparison between XMS, MADEUS and XADAPTOR according to adaptation approaches factors discussed in Section 1.

An extensive comparison between XMS and other existing solutions that tackle the problem of adaptation is presented in Table 9, the compared solutions include the previously closely similar solutions to XMS (MADEUS, XADAPTOR) (see Table 8).

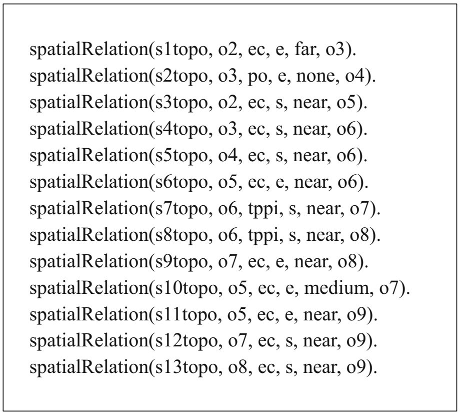
In the comparison table we used the following criteria and questions that an adaptation solution might address:

- Adaptation of object: this is a primary criteria in developing a solution for adaptation, the authors of the solution choose whether adapting single media elements is of concern for the solution under development or not.



```
?- time(perceptTopologyRefine(reduce,d1,smartPhone,X)).
% 166 inferences, 0.000 CPU in 0.000 seconds (?% CPU, Infinite Lips)
X = d1toporefine .
```

Fig. 28 Topology refining predicate call in PROLOG



```
spatialRelation(s1topo, o2, ec, e, far, o3).
spatialRelation(s2topo, o3, po, e, none, o4).
spatialRelation(s3topo, o2, ec, s, near, o5).
spatialRelation(s4topo, o3, ec, s, near, o6).
spatialRelation(s5topo, o4, ec, s, near, o6).
spatialRelation(s6topo, o5, ec, e, near, o6).
spatialRelation(s7topo, o6, tppi, s, near, o7).
spatialRelation(s8topo, o6, tppi, s, near, o8).
spatialRelation(s9topo, o7, ec, e, near, o8).
spatialRelation(s10topo, o5, ec, e, medium, o7).
spatialRelation(s11topo, o5, ec, e, near, o9).
spatialRelation(s12topo, o7, ec, s, near, o9).
spatialRelation(s13topo, o8, ec, s, near, o9).
```

Fig. 29 Spatial relations refined after topology refining call

- Adaptation of composition: same as adapting objects, the authors of an adaptation solution should choose if the solution require adapting the whole multimedia composition or simply media elements.
- Alternative specification: this criteria defines if the solution proposes a way to define some kind of alternatives whether to a particular object within the composition, or for the entire document. The ‘author’, ‘user’ values defines the case where the alternatives are defined by the author or the user.
- Spatial specification: this defines if the solution propose a way to specify the spatial dimension of the document.
- Temporal specification: this defines if the solution propose a way to specify the temporal dimension of the document.
- Hypermedia specification: this defines if the solution propose a way to specify the hypermedia dimension of the document.
- Separation: this criteria defines where applicable, if there is a separation between the content and the specification.
- Transformation rules: defines whether the solution proposes a way to define or has predefined rules for transforming either objects or the whole composition.
- Flexible document: defines if the solution is based on a flexible document model, in the case of XMS ‘author’ means its up to the author to decide the level of flexibility of the specification.
- Replacement: defines whether the solution is based on a document that replaces the original one.
- For existing document: defines if the solution is used on existing documents, or is based on a specific new document defined as a part of the solution.

Table 8 Comparison between XMS, MADEUS and XADAPTOR

Factor	XMS	MADEUS	XADAPTOR
Abstraction	1-2	0	0
Workload	1-2	2	1
Separation	2	0	0
Flexibility	1-2	1	1
Author control	1-2	0	0

Table 9 Comparison between XMS and other similar solutions

Solution	XMS	MADEUS [12]	XADAPTOR [20]	WebViews [15]	AHM [19]	MPEG 21 [6]	SMIL [4]	AHA [9]	Cuypers [18]	MM4U [37]
Adaptation of object	No	No	No	No	No	Yes	No	No	No	No
Adaptation of composition	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes
Alternative specification	Author	No	No	User	Yes	Yes	Yes	No	No	No
Spatial specification	Yes	Yes	No	No	No	No	Yes	No	Yes	Yes
Temporal specification	Yes	Yes	No	No	No	No	Yes	No	Yes	Yes
Hypermedia specification	No	Yes	No	No	Yes	No	Yes	Yes	Yes	No
Separation	Yes	No	N/A	No	No	N/A	No	N/A	No	No
Transformation rules	Yes	Yes	Yes	No	No	No	No	Yes	No	No
Flexible document	Author	No	No	No	No	No	No	No	Yes	Yes
Replacement	No	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes
For existing documents	Yes	No	Yes	Yes	No	No	No	Yes	No	No
Level of specification	Author	Pre-defined	Pre-defined	Pre-defined	Pre-defined	Pre-defined	Pre-defined	Pre-defined	Pre-defined	Pre-defined
Level of flexibility	Author	Pre-defined	Pre-defined	Pre-defined	Pre-defined	Pre-defined	Pre-defined	Pre-defined	Pre-defined	Pre-defined

- Level of specification: the level of details included in the specification, XMS leaves this to the sole decision of the author of the document and the specification.
- Level of flexibility: the level of which the solution provide flexibility in the adaptation, in all the solutions the level is predefined except XMS where the flexibility is up to the author of the document.

7 Conclusion

The aim of our work was to propose a generic approach for profile based multimedia document adaptation. It uses inference systems to modify a multimedia specification according to a given profile, we ended up by proposing a novel abstract document processing model. It is based on an abstract document called XMS that describes and accompany an original multimedia document, XMS is a replacement in the processing model to the original document and allows to generate a new adapted version using a set of manipulations to modify its spatial dimension.

We focused on the spatial dimension of a specification, a combination of spatial qualitative information were used to describe the spatial composition of the original multimedia document. Qualitative topologies, directions and distances were combined into a composite spatial relation to describe the relation between each two objects of a specification.

The key contributions of the approach based on XMS can be summed up as follows:

- The use of XMS as an abstract replacement of the original document in the processing model allowed us to achieve the goals set in the introduction, this also helped to maintain the separation between the original document and the adaptation system.
- The combination of multiple spatial qualitative information allows an accurate definition of the spatial layout; the proposed reasoning helps not only reduce the space occupied by the spatial specification but also extend it (or stretch) to fit a bigger output or simply a bigger percept.
- Three basic algorithms for manipulating distances, directions and topologies were proposed. The algorithms implemented in PROLOG forms a basis for the main algorithm of the adaptation, further work is needed and investigation about all the possibilities of possible manipulations that help achieve the goal by balancing limitations, authors intentions, and user preferences.
- The main algorithm assembles the basic algorithms, process, and test if the resulted specification matches a profile. In some cases, only basic distances refining may adapt the specification if the profile is for a situation not so much different than the original intended execution situation.
- The combined qualitative spatial information and the manipulation algorithms proposed in this paper may have a broader applicability in other fields other than multimedia. Such fields may include and not limited to the following:
    - Visual tracking problems, further readings in [25–27].
    - Geography and geophysics problems, further readins in [14, 22, 34].
    - Pattern recognition problems, further readings in [8, 28, 39].

The reasoning in its current version does not exploit all the advantages provided in XMS, given that XMS also proposes ways to define limitations over the adaptation such as restrictions, alternatives, and also rule based alternatives.

Another limitation of the current version is it only processes the spatial dimension of the specification, a future work including temporal and even hypermedia dimensions is under study. A similar approach to spatial reasoning can be used on the temporal dimension using ALLEN interval logic rules [2, 12, 16].

A challenge would be to provide a full framework for multimedia document adaptation, using both spatio-temporal and hypermedia reasoning combined adaptation based on the specification given by the author using XMS. Giving the author the full control on his document while allowing much of the flexibility while processing the specification through XMS.

References

1. Adel A, Laborie S, Roose P (2014) Semantic context-aware adaptation platform architecture. Procedia Comput Sci 32:959–964
2. Allen JF (1983) Maintaining knowledge about temporal intervals. Commun ACM 26(11):832–843
3. Archana P, Kulkarni SS (2015) Meta-heuristic approach for resource optimization in mobile real time video traffic. Int J Comput Appl 110:14
4. Bulterman DC, Rutledge LW (2008) SMIL 3.0: flexible multimedia for web mobile devices and daisy talking books. Springer Publishing Company Incorporated
5. Burnett I, Van de Walle R, Hill K, Bormans J, Pereira F (2003) Mpeg-21: goals and achievements. IEEE MultiMed 10(4):60–70
6. Burnett IS, Pereira F, Van de Walle R, Koenen R (2006) The MPEG-21 book. Wiley
7. Cavallaro A (2005) Universal multimedia access. In: Encyclopedia of multimedia technology and networking. IGI Global, pp 1001–1007
8. Chuang KS, Tzeng HL, Chen S, Wu J, Chen TJ (2006) Fuzzy c-means clustering with spatial information for image segmentation. Comput Medical Imag Graph 30(1):9–15
9. De Bra P, Smits D, Stash N (2006) The design of aha! In: Proceedings of the seventeenth conference on hypertext and hypermedia. ACM, pp 133–134
10. Dromzée C, Laborie S, Roose P (2013) A semantic generic profile for multimedia documents adaptation. In: Intelligent Multimedia technologies for networking applications: techniques and tools, pp 225–246
11. Ellis TJ (2001) Multimedia enhanced educational products as a tool to promote critical thinking in adult students. J Educ Multimed Hypermed 10(2):107–124
12. Euzenat J, Layaïda N, Dias V (2003) A semantic framework for multimedia document adaptation. In: Proceedings of the 18th international joint conference on artificial intelligence IJCAI'2003. Morgan Kauffman, Acapulco, pp 31–36. https://hal.inria.fr/inria-00423411
13. Frank AU (1992) Qualitative spatial reasoning about distances and directions in geographic space. J Vis Lang Comput 3(4):343–371
14. Frank AU (1996) Qualitative spatial reasoning: cardinal directions as an example. Int J Geogr Inf Sci 10(3):269–290
15. Freire J, Kumar B, Lieuwen D (2001) Webviews: accessing personalized web content and services. In: Proceedings of the 10th international conference on World Wide Web. ACM, pp 576–586
16. Gerevini A, Nebel B (2002) Qualitative spatio-temporal reasoning with rcc-8 and allen's interval calculus: computational complexity. In: Proceedings of the 15th European conference on artificial intelligence. IOS Press, pp 312–316
17. Gerevini A, Renz J (2002) Combining topological and size information for spatial reasoning. Artif Intell 137(1–2):1–42
18. Geurts J, van Ossenbruggen JR, Hardman HL (2001) Application-specific constraints for multimedia presentation generation. Centrum voor Wiskunde en Informatica
19. Hardman L, Bulterman DC, Van Rossum G (1994) The amsterdam hypermedia model: adding time and context to the Dexter model. Commun ACM 37(2):50–62
20. He J, Gao T, Hao W, l Yen I, Bastani F (2007) A flexible content adaptation system using a rule-based approach. IEEE Trans Knowl Data Eng 19(1):127–140. https://doi.org/10.1109/TKDE.2007.250590
21. Held A, Buchholz S, Schill A (2002) Modeling of context information for pervasive computing applications. In: Proceedings of SCI, pp 167–180

22. Jones CB, Purves R, Ruas A, Sanderson M, Sester M, Van Kreveld M, Weibel R (2002) Spatial information retrieval and geographical ontologies an overview of the spirit project. In: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 387–388
23. Katai Z, Toth L (2010) Technologically and artistically enhanced multi-sensory computer-programming education. Teach Teacher Educ 26(2):244–251
24. Kiss C (2007) Composite capability/preference profiles (cc/pp): structure and vocabularies 2.0. W3C Working Draft, p 30
25. Lan X, Ma AJ, Yuen PC (2014) Multi-cue visual tracking using robust feature-level fusion based on joint sparse representation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1194–1201
26. Lan X, Zhang S, Yuen PC (2016) Robust joint discriminative feature learning for visual tracking. In: IJCAI, pp 3403–3410
27. Lan X, Zhang S, Yuen PC, Chellappa R (2018) Learning common and feature-specific patterns: a novel multiple-sparse-representation-based tracker. IEEE Trans Image Process 27(4):2022–2037
28. Lasch P, Naumann D (1998) Ft-ir microspectroscopic imaging of human carcinoma thin sections based on pattern recognition techniques. Cellular Molec Biol (Noisy-le-Grand France) 44(1):189–202
29. Lei Z, Georganas ND (2001) Context-based media adaptation in pervasive computin. In: Canadian Conference on electrical and computer engineering, 2001, vol 2. IEEE, pp 913–918
30. Li Q, Yang J, Zhuang Y (2008) Multimedia information retrieval at a crossroad. In: Multimedia technologies: concepts, methodologies, tools, and applications. IGI Global, pp 242–249
31. Louafi H, Coulombe S, Chandra U (2015) Robust qoe-aware prediction-based dynamic content adaptation framework applied to slides documents in mobile web conferencing. Multimed Tools Appl 74(18):7883–7920
32. Mohan R, Smith JR, Li CS (1999) Adapting multimedia internet content for universal access. IEEE Trans Multimed 1(1):104–114
33. Osbakk P, Ryan NS (2002) Context, cc/pp, and p3p. UbiComp 2002 adjunct proceedings, pp 9–10
34. Renz J (2002) Qualitative spatial reasoning with topological information. Springer-Verlag
35. Reynolds F (1999) Composite capability/preference profiles (cc/pp): a user side framework for content negotiation. W3C note
36. Saravanaselvam N, Rajangam K, Dhanam L, Balasubramanian A (2015) Mobile web browsing based on content preserving with reduced cost. Int J Adv Network Appl 6(4):2398
37. Scherp A, Boll S (2008) Mm4u: A framework for creating personalized multimedia content. In: Multimedia technologies: concepts, methodologies, tools, and applications. IGI Global, pp 325–363
38. Sharma J, Flewelling DM (1995) Inferences from combined knowledge about topology and directions. In: International Symposium on spatial databases. Springer, pp 279–291
39. Van Essen DC, Anderson CH, Felleman DJ (1992) Information processing in the primate visual system: an integrated systems perspective. Science 255(5043):419–423
40. Vetro A (2004) Mpeg-21 digital item adaptation: enabling universal multimedia access. IEEE Multimed 11(1):84–87
41. Wang Y, Kim JG, Chang SF, Kim HM (2007) Utility-based video adaptation for universal multimedia access (uma) and content-based utility function prediction for real-time video transcoding. IEEE Trans Multimed 9(2):213–220
42. Yuan Z, Ghinea G, Muntean GM (2015) Beyond multimedia adaptation: quality of experience-aware multi-sensorial media delivery. IEEE Trans Multimed 17(1):104–117
43. Zhang D, Jangam A, Zhou L, Yakut I (2015) Context-aware multimedia content adaptation for mobile web. Int J Netw Distrib Comput 3(1):1–10

Yehya Belhadad got his master of science from the department of computer science in the faculty of Sciences - University of Ferhat Abbes Sétif 1/Algeria, part of a research team in the laboratory of networks and distributed systems L.R.S.D of the same university. Currently a Ph.D. student also in the same university. His main topics of interrest are Natural language processing, PROLOG, optimization, multimedia adaption, context-aware, multimedia documents, software architecture and development, artificial intelligence and knowledge based systems.

Allaoua Refoufi is an associate Professor at the department of computer science in the faculty of Sciences - University of Ferhat Abbes Sétif 1 /Algeria, head of a research team in the laboratory of networks and distributed systems L.R.S.D of the same university. He supervised multiple PhD thesis and was also involved in several PhD defenses. He published many national and international articles. His works are mainly focuses on Natural language processing, PROLOG, optimization, multimedia adaption, context-aware, ontologies and multimedia documents.

Philippe Roose is an associate Professor at LIUPPA research lab - University of Pau/France. Currently he is director of the computer science dept. and head of the T2I Research Team involving 17 permanents. He is co-author of the Kalimucho TM platform (patented). He supervised 9 PhD thesis and was involved in more than 30 PhD defenses. He published many national and international articles, journals and books. His works are mainly focuses on middleware, software architecture, dynamic adaption, context-aware, and multimedia documents.