Journal of Engineering Science and Technology Review 15 (3) (2022) 170 - 176

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Speedup Multimedia Authoring Process: Incrementality of Multimedia Authoring Verification

Marvin Chandra Wijaya[1,2*], Zulisman Maksom[1] and Mohammad Haziq Lim Abdullah[1]

[1]*Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Melaka 76100, Malaysia*
[2]*Departement of Computer Engineering, Maranatha Christian University, Bandung 40164, Indonesia*

Received 27 July 2021; Accepted 16 July 2022

Abstract

Multimedia authoring tool performs to combine various kinds of media, both visual and non-visual media into a multimedia document. The process for converting media objects into multimedia documents in the form of multimedia programming uses a mandatory process called the kernel mechanism. The kernel mechanism consists of editing, verification, modeling and translation processes. In the kernel mechanism there is a time-consuming verification process. Verification used in the kernel mechanism is time computation verification and Spatio-temporal verification. With the increasing number of media objects, the verification process time will be longer. A novel method proposed in this study is incrementality of time computation and incrementality of spatio-temporal verification. The Incrementality process uses a specific algorithm to process verification on media objects that have been modified by the author. Experiments that have been carried out in this study show that this method is successful in speeding up the verification process significantly.

Keywords: Multimedia Auhtoring, Verifications, Incrementality

1. Introduction

A multimedia authoring tool is an application that has a function to combine various kinds of media objects, both visual media, and non-visual media [1], [2]. Multimedia authoring tool will produce a multimedia document that can be a multimedia language [3]. The existing multimedia programming languages include Synchronized Multimedia Integration Language (SMIL), Nested Context Language (NCL), and HTML5 [4], [5]. In a multimedia authoring tool, there are several steps must be done. These steps are editing, verification, modeling, and translation. At first, the author must enter the data of media objects into the multimedia authoring tools [6]. In this editing process, the author enters data in the form of temporal layout and spatial layout, as shown in Fig 1 and Fig 2 [7], [8].

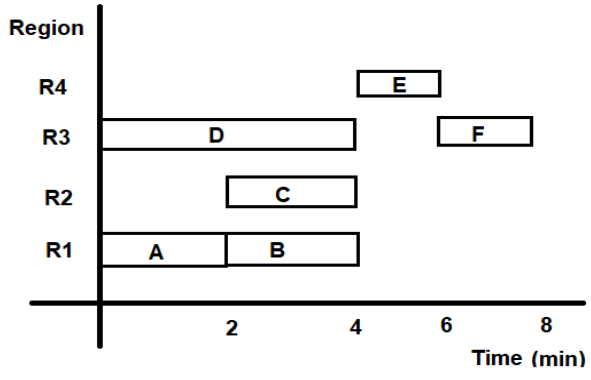

Fig. 1. Temporal Layout

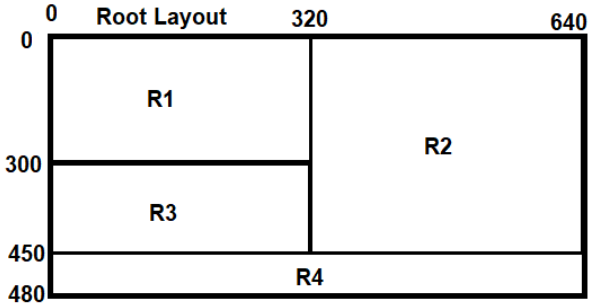

Fig. 2. Spatial Layout

A multimedia authoring tool must help the author reduce errors that may occur, so a verification process is needed to check for any errors. Verification is a process that requires a lot of processing time. On verification, the time required to process the verification algorithm is highly dependent on the number of media objects in a multimedia presentation. The more media objects in a multimedia document, the longer the processing time will be. The relationship between the processing time required and the number of objects is exponential, as shown in Fig 3.

An author usually performs the verification process many times during the creation of a multimedia presentation, and this causes the design of a multimedia presentation to be very time-consuming. Even though there are only slight changes in the multimedia presentation during the repetition of the process, it still takes the same time as repeating the whole process. Verification methods that have been widely used for multimedia authoring tools are time computation and Spatio-temporal [9]–[11]. In this study, a novel method is proposed to improve existing verification methods to speed up the verification process. The verification method was modified into incrementality of time computation and incrementality of

*E-mail address: marvinchw@gmail.com
ISSN: 1791-2377
doi:10.25103/jestr.153.19

Spatio-temporal verification.

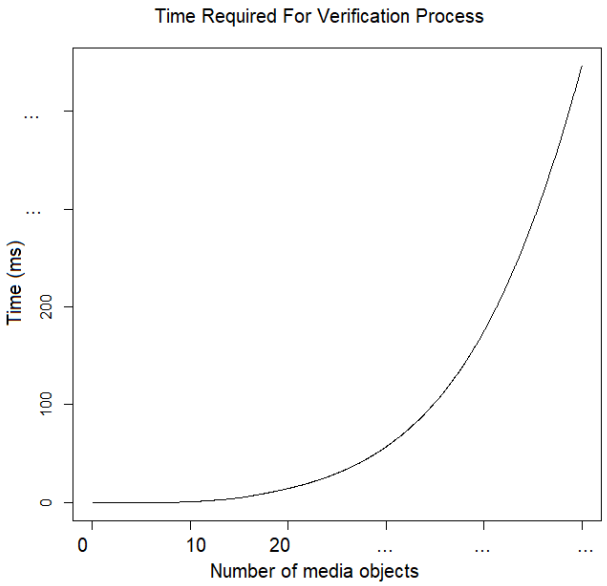

Fig. 3. Time required for verification process

2. Kernel Multimedia Authoring Mechanism

The kernel multimedia authoring mechanism is a series of processes that must be implemented in a multimedia authoring tool. The steps in the kernel mechanism are shown in Fig 4 [12].

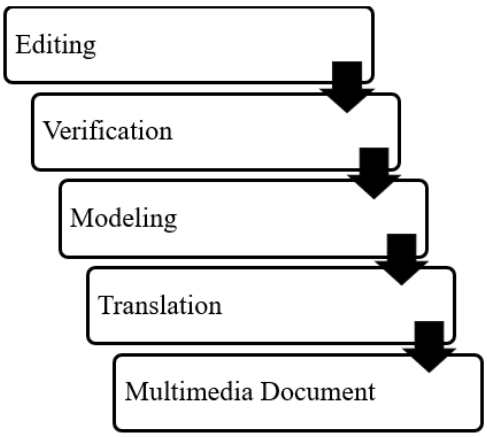

Fig. 4. Kernel Mechanism

The kernel mechanism is a mandatory process carried out by the multimedia authoring tool to produce multimedia documents. Several issues regarding multimedia documents must be considered in making multimedia presentations: making multimedia languages, including the development of conceptual models, design of media settings, and variations of media appearance.

There are five main things in the kernel mechanism: Editing, Verification, Modeling, Translation, and Multimedia Document. The editing process involves the author and the interface of multimedia authoring, while the other four processes are automatic processes according to their respective algorithms. The verification process usually consists of the verification of the temporal layout and spatial layout. Verification that is widely used in multimedia authoring tools is time computation verification and Spatio-temporal verification. Verification is an essential step in the kernel mechanism of the multimedia authoring tool because the next process (i.e., modeling) can be processed if the verification is carried out correctly[13]. The multimedia authoring model mainly uses several models as follows: Petri Net, Hoare Logic, Simple Interactive Multimedia Model (SIMM), and Language of Temporal Ordering Specifications (LOTOS) [1], [14]–[19]. A model of a multimedia presentation can be translated into a multimedia document with a specific algorithm. A multimedia document correctly generated by a multimedia authoring tool must be playable by a multimedia player [20].

2.1. Time Computation Verification

The timing computation verification calculates the overall time duration. If there is a change in time, it is done gradually from the low level first. If there are no errors at the low level, the calculation will proceed to a higher level. But if it turns out there is an error at a level/group of elements/media, the calculation and verification process will be stopped first and give an error warning to the author.

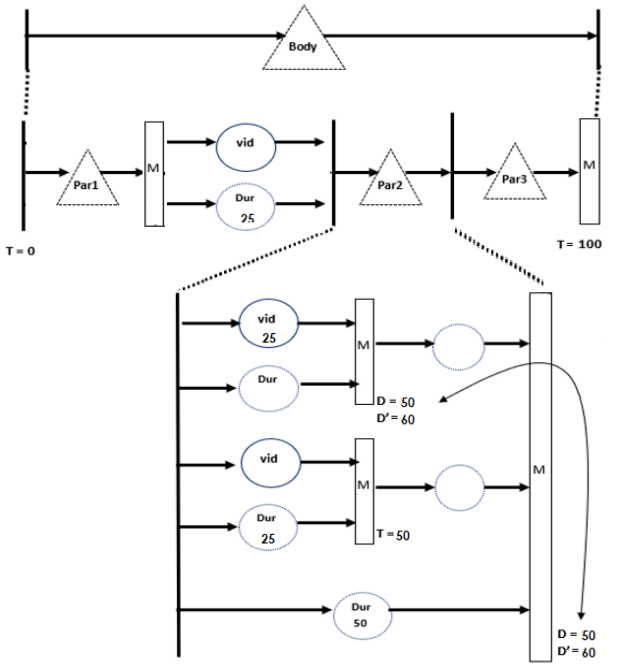

Fig. 5. Time Computation Verification

Fig 5. is an example of calculating time computation verification in a multimedia presentation. The calculation of time computation verification uses the Petri Net model, where there is one sequential group in which there is a parallel-group. The time duration in each media object is calculated to verify whether there is an error or not.

2.2. Spatio-temporal verification

Visual media objects such as images, videos, animations, and text that appear in the timeline can overlap in time and spatial layout [21]. As shown in Fig. 6, there are two types of spatial conflicts: the entire spatial conflict in two regions (region R1 and region R2) and partial spatial conflict in two regions

(region R1 and region R2).

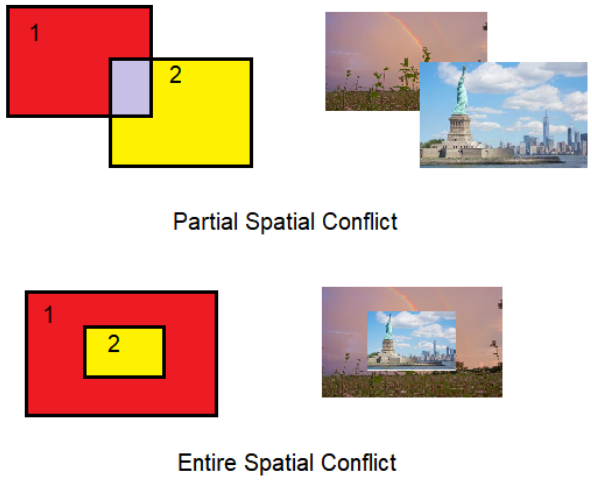

Fig. 6. Spatial Conflict

2.3. Diagnosis Service

Diagnosis Service is a Multimedia Authoring tool that can provide error information that occurs to the author. Error information given to the author is in the form of line location information in the multimedia programming code that experiences errors or conflicts. The information provided also contains error information that occurred at the location of the line. Usually, the user interface has a special window or pop window to provide information to the author.

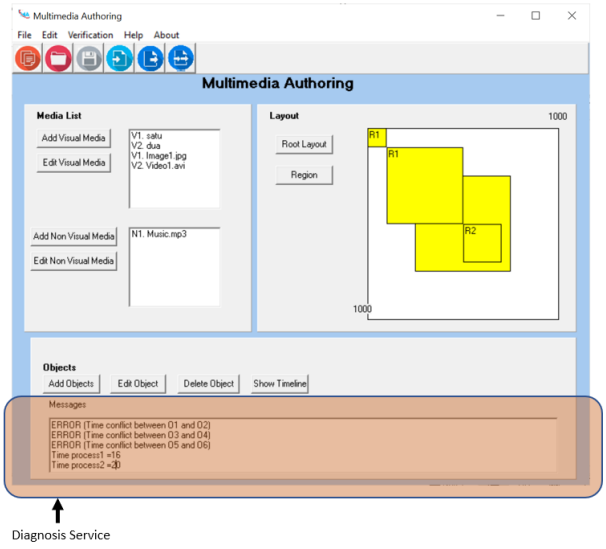

Fig. 7. Diagnosis Service

Fig 7. shows an example of a diagnostic service in a multimedia authoring tool. At the bottom of the interface, there is a box containing a message for the author. The message contains error information that occurs when editing a multimedia presentation. The error information that appears in the box contains the types of errors, their location, and when the error occurred.

3. Method

The time it takes to process a multimedia presentation depends on the number of media objects and regions. Most of the time is needed to carry out the verification process (time computation verification and Spatio-temporal verification). The more media objects in a multimedia presentation will require a longer processing time. A novel method proposed in this study is to use incrementality for both verifications, as shown in Fig 8.

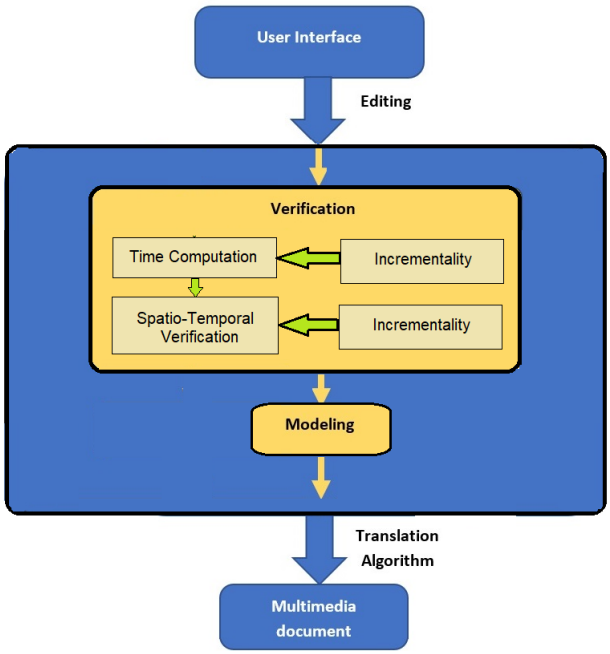

Fig. 8. A novel verification Method using incrementality

3.1. Incrementality

Incrementality is the ability to process data that considers changes in input data [22]. The concept of Incremental is the ability of the software to recalculate only the changed input data. This method will save processing time [13]. By conducting incremental modeling and verification, it is not necessary to parse the models that have been made. In order to create a system that can detect errors as quickly as possible during the editing process, a system with a model that has incremental verification is needed.

Modeling called the SMIL Net hierarchy (H SMIL Net) can carry out incremental verification. Systems that can do incrementality verification have the advantage of having a good response time. The model has a temporal structure that is suitable for incrementality verification. Graphical representation for body place, sequence (seq) place, and parallel (par) place are shown in Fig. 9, and examples of Hierarchy SMIL Petri Net (H-SMIL Net) are shown in Fig 10.

Fig. 9. Graphical representation of composite places

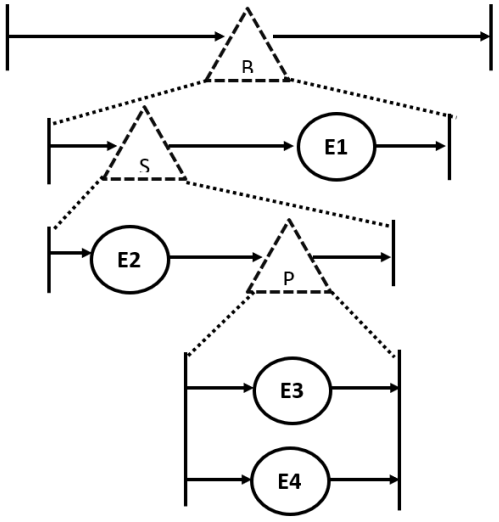

Fig. 10. Hierarchy SMIL Petri Net (H-SMIL Net) example

3.2. Incrementality of Time Computation Verification

Incrementality of time computation verification is used to speed up the verification process. The time computation verification algorithm will be applied to every object in a multimedia presentation in the initial conditions or have never been verified. Meanwhile, if the author changes the objects in a multimedia presentation, the time computation verification algorithm will be applied to the objects modified by the author. The verification process on each object will be divided for each region in a multimedia presentation. In each region, verification will be carried out on each object to compare the start time and end time. If there is time overlap for objects in the same region, the author will be given an error message.

The formula for checking time conflicts in the same region is as follows:

$\forall$ $Object_i$, $\forall$ $Object_j$, $i < n$, $j < n$, $\exists$ $End(Object_i) < Begin(Object_j)$

Where:
$Object_i$ = Media $Object_i$
$Object_j$ = Media $Object_j$
n = number of objects
End = End time
Begin = Begin time

The designed time computation verification algorithm is as follows:

1. Calculation of the number of objects for each region
2. Changing the shape of a 2-dimensional timeline array into a 3-dimensional array
3. Sort objects by time begin
4. While regions
5. If an object is changed
6. While object
7. IF Object_end < Previous Object_begin then notify the author
8. end while
9. end if
10. end while

3.3. Incrementality of Spatio-temporal Verification

Spatio-temporal verification verifies objects in different overlapping regions. Each region in a multimedia presentation will be checked first whether there is overlapping, either spatial overlapping or entire overlapping. The next process is to list all possible overlaps in all regions. After obtaining a list of overlapping regions, all objects in the list are checked as in time computation verification. The verification is not applied to objects with the same region, but to objects with overlapping regions. In the Incrementality of Spatio-temporal verification, the verification will be applied to objects that the author has changed.

Two regions will be considered as overlapping regions if they fulfill the following Boolean equations:

((L2>L1) && (L2<L1+W1)) || ((L2+W2>L1) && (L2+W2<L1+W1)) && ((T2>T1) && T2<T1+H1)) || ((T2+H2>T1) && (T2+H2<T1+H1))

Where:
L1,T1 = left top position of object 1
L2, T2 = left top position of object 2
W1, H1 = width height of object 1
W2, H2 = width height of object 2

The designed Spatio-temporal verification algorithm is as follows:

1. Calculation number of objects for each region
2. Changing the shape of a 2-dimensional timeline array into a 3-dimensional array
3. Sort objects by time begin
4. While regions
5. Calculation of overlapping regions
6. End while
7. While overlapping_regions{
8. If an object is changed
9. While first_regions{
10. While second_regions{
11. If first_regions_object begin < second_regions_object begin < first_regions_object end then notify the user
12. End while
13. End While
14. End If
15. End While

4. Discussion and Results

The verification algorithm that has been designed in the previous section experimented with several simple multimedia presentation scenarios. The experimental results are tested whether the results have been running properly and analyzed for speed up during the verification process. Also carried out a comparison of the processing time required for verification by using and without using the incrementality algorithm.

Example of temporal verification processes using time computation with Petri Net model are as follows:

```
<smil>
<head>
 <layout>
```

```
 <root-layout width="800" height="800"/>
 <region id="reg1" width="400" height="400" top="0" left="0"/>
 <region id="reg2" width="400" height="400" top="0" left="400"/>
 <region id="reg3" width="400" height="400" top="400" left="0"/>
 <region id="reg4" width="400" height="400" top="400" left="400"/>
 </layout>
</head>
<body>
 <seq>
  <video src="vid1.mp4" id="img1" region="reg1" dur="50"/>
  <par dur="50s">
  <video src="vid1.mp4" id="img1" region="reg2" dur="50"/>
  <video src="vid2.mp4" id="img1" region="reg3" dur="50"/>
  </par>
  <par dur="50s">
  <video src="vid3.mp4" id="img1" region="reg2" dur="50"/>
  <seq dur=”50s”>
   <video src="vid4.mp4" id="img1" region="reg3" dur="25"/>
   <video src="vid5.mp4" id="img1" region="reg4" dur="25"/>
  </seq>
  </par>
  <par dur="50s">
   <video src="vid6.mp4" id="img1" region="reg2" dur="50"/>
   <video src="vid7mp4" id="img1" region="reg3" dur="50"/>
   <video src="vid8.mp4" id="img1" region="reg3" dur="50"/>
  </par>
 </seq>
</body>
</smil>
```

If the duration of the “vid1” element is changed from 50 to 60, the algorithm for detecting the change is:

- Step1 : D (par1) = 100
  D '(par1) = 110
- Step 2: D (par1) = D (par1)
- Step 3 :"par 1" is invariant

Where :

D = Total duration before modification
D’ = Total duration after modification

From the process of the algorithm, it was found that the value of D = 100, which is invariant (because it is explicitly written in a multimedia document), turns out to have a different value from the results of the calculation of time. This change causes the Kernel Mechanism to detect errors and stop the process further. The User Interface can display to the author that a time conflict has occurred.

Table 1. Comparison of the time required for the time computation verification process

Number of Objects	Number of Region	Number Objects per region	Time Required (without Incrementality)	Time Required (with Incrementality)
20	10	2	21 ms	4 ms
20	2	10	39 ms	6 ms
40	10	4	79 ms	13 ms
40	4	10	77 ms	12 ms
60	10	6	120 ms	22 ms
60	6	10	114 ms	20 ms
80	10	8	158 ms	27 ms
80	8	10	149 ms	25 ms
100	10	10	197 ms	35 ms

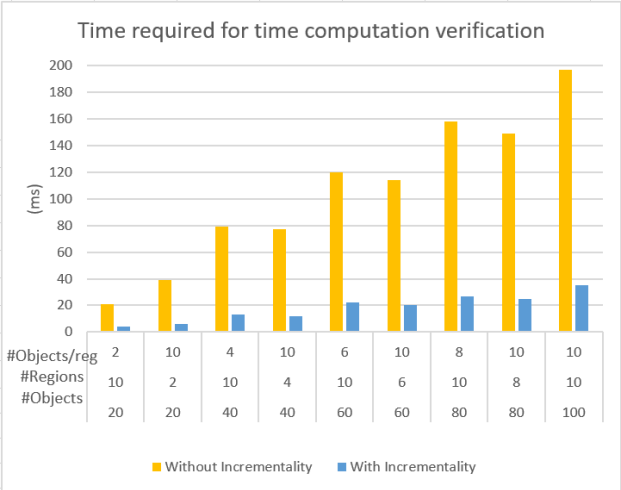

Fig. 11. Time required for time computation verification

Table 1 and Fig 11. compares the time required for the time computation verification process using and without using the incrementality algorithm. Based on the experimental results, it was found that if the author changes about 10% of objects from a multimedia presentation, it will significantly reduce the verification process time.

Spatial verification detects overlapping spatial at certain time intervals and looks for empty areas that will not cause spatial conflicts. As an example illustrated in the program, the following causes:

```
<smil>
<head>
 <layout>
  <root-layout width="800" height="800"/>
  <region id="reg1" width="400" height="300" top="0" left="100"/>
  <region id="reg2" width="400" height="300" top="500" left="0"/>
  <region id="reg3" width="200" height="200" top="600" left="600"/>
  <region id="reg4" width="300" height="400" top="200" left="200"/>
 </layout>
</head>
<body>
 <par>
  <video src="vid1.mp4" id="img1" region="reg1" dur="50"/>
  <video src="vid2.mp4" id="img2" region="reg2" dur="50"/>
  <video src="vid3.mp4" id="img3" region="reg3" dur="50"/>
  <video src="vid3.mp4" id="img4" region="reg4" dur="50"/>
 </par>
</body>
</smil>
```

Fig 12. shows the processing results of an algorithm that detects overlap between several regions. The experimental

results in the program managed to get two overlaps between region 1 and region 4 as well as overlapping between region 2 and region 4.

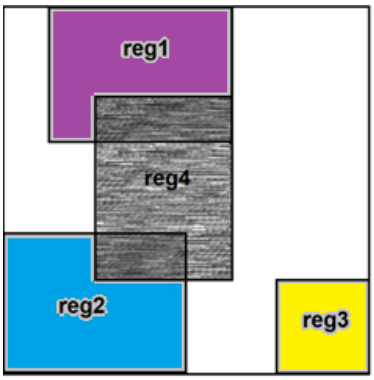

Fig. 12. Spatial object overlapping

Table 2. Comparison of the time required for the Spatio-temporal verification process

Number of Objects	Number of Region	Number Objects per region	Time Required (without Incrementality)	Time Required (with Incrementality)
20	10	2	297 ms	58 ms
20	2	10	102 ms	21 ms
40	10	4	860 ms	160 ms
40	4	10	627 ms	110 ms
60	10	6	1726 ms	299 ms
60	6	10	1676 ms	280 ms
80	10	8	2999 ms	574 ms
80	8	10	2953 ms	570 ms
100	10	10	4668 ms	810 ms

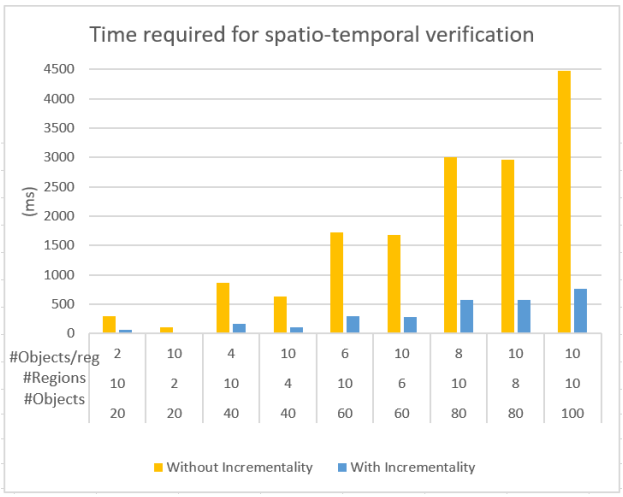

Fig. 13. Spatial object overlapping

Table 2 and Figure 13. compare the time required for the Spatio-temporal verification process using and without using the incrementality algorithm. In this type of verification, it is also found that if the author changes about 10% of objects from the multimedia presentation, it also will significantly reduce the verification process time.

5. Conclusions

A multimedia authoring tool performs to convert a multimedia presentation into a multimedia document in the form of multimedia programming. A kernel mechanism is a mandatory process in a multimedia authoring tool, namely editing, verification, modeling, and translation. Of all these processes, the verification process is time-consuming. The time required for the verification process is highly dependent on the number of objects in a multimedia presentation. Speed up the verification process is needed by an author in designing multimedia presentations. A novel method for the verification algorithm that functions to speed up verification time, the method takes the form of incrementality time computation and incrementality Spatio-temporal verification.

Incrementality is the ability to process data considering changes in input data. The Incremental concept is the ability of the software to recalculate only the changed input data to save processing time. Experimental results on incrementality verification show significant results in speed up verification time both on time computation and Spatio-temporal verification.

References

1. De Mattos, D. P., Muchaluat-Saade, D. C. Steve: A hypermedia authoring tool based on the simple interactive multimedia model. Proceedings of the ACM Symposium on Document Engineering 2018, DocEng 2018 14(4) (2018) 1–24. https://10.1145/3209280.3209521.
2. Belhadad, Y., Refoufi, A., Roose, P. Spatial reasoning about multimedia document for a profile based adaptation: Combining distances, directions and topologies. Multimedia Tools and Applications 77(23) (2018) 30437–30474. https://10.1007/s11042-018-6080-8.
3. Wijaya, M. C., Maksom, Z., Abdullah, M. H. L. A Brief of Review : Multimedia Authoring Tool Attributes. Ingénierie des Systèmes d'Information 26(1) (2021) 1–11. https://10.18280/isi.260101.
4. Guimarães, R. L., Cesar, P., Bulterman, D. C. A. Creating and sharing personalized time-based annotations of videos on the web. DocEng2010 - Proceedings of the 2010 ACM Symposium on

Document Engineering (2010) 27–36. https://10.1145/1860559.1860567.

5. Jourdan, M. A formal semantics of SMIL: A web standard to describe multimedia documents. Computer Standards and Interfaces 23(5) (2001) 439–455. https://10.1016/S0920-5489(01)00091-5.
6. Huang, B., Yi, S., Chan, W. T. Spatio-temporal information integration in XML. Future Generation Computer Systems 20(7) (2004) 1157–1170. https://10.1016/j.future.2003.11.005.
7. Yang, C. C., Yang, Y. Z. SMILAuthor: An Authoring System for SMIL-Based Multimedia Presentations. Multimedia Tools and Applications 21(3) (2003) 243–260. https://10.1023/A:1025770817293.
8. Yang, C., Chu, C.-K., Wang, Y.-C. Extension of Timeline-based Editing for Non-deterministic. Journal Of Information Science and Engineering 24(5) (2008) 1377–1395. .
9. Wijaya, M. C., Maksom, Z., Abdullah, M. H. L. Two Verification Phases in Multimedia Authoring Modeling. Journal of information and communication convergence engineering 19(1) (2021) 42–47. https://10.6109/jicce.2021.19.1.42.
10. Bouyakoub, S., Belkhir, A. A Spatio-Temporal Authoring Tool for Multimedia SMIL Documents. International Journal of Multimedia Technologies 2(3) (2012) 83–88. .
11. Bai, L., Yan, L., Ma, Z. M., Xu, C. Incorporating fuzziness in spatiotemporal XML and transforming fuzzy spatiotemporal data from XML to relational databases. Applied Intelligence 43(4) (2015) 707–721. https://10.1007/s10489-015-0677-7.
12. Bouyakoub, S., Belkhir, A. SMIL builder: An incremental authoring tool for SMIL documents. ACM Transactions on Multimedia Computing, Communications and Applications 7(1) (2011) 1–30. https://10.1145/1870121.1870123.
13. Júnior, D. P., Farines, J. M., Koliver, C. (2012). , An approach to verify live NCL applications. , in *WebMedia'12 - Proceedings of the 2012 Brazilian Symposium on Multimedia and the Web*, pp. 223–231., (2012). , https://10.1145/2382636.2382685.
14. Bouyakoub, S., Belkhir, A. (2008). , H-SMIL-Net: A hierarchical Petri Net model for SMIL documents. , in *Proceedings - UKSim 10th International Conference on Computer Modelling and Simulation, EUROSIM/UKSim2008*, pp. 106–111., (2008). , https://10.1109/UKSIM.2008.54.
15. Mekahlia, F. Z., Ghomari, A., Yazid, S., Djenouri, D. Temporal and Spatial Coherence Verification in SMIL Documents with Hoare Logic and Disjunctive Constraints: A Hybrid Formal Method. Journal of Integrated Design and Process Science 20(3) (2017) 39–70. https://10.3233/jid-2016-0020.
16. Sampaio, P. N. M., Courtiat, J.-P. An approach for the automatic generation of RT-LOTOS specifications from SMIL 2.0 documents. Journal of the Brazilian Computer Society 9(3) (2004) 39–51. https://10.1590/s0104-65002004000100004.
17. Léonard, L., Leduc, G. An introduction to ET-LOTOS itive systemsfor the description of time-sens. Computer Networks 29(3) (1997) 271–292. .
18. Logrippo, L., Faci, M., Haj-Hussein, M. An introduction to LOTOS: learning by examples. Computer Networks and ISDN Systems 23(5) (1992) 325–342. https://10.1016/0169-7552(92)90011-E.
19. Courtiat, J. P., Santos, C. A. S., Lohr, C., Outtaj, B. Experience with RT-LOTOS, a temporal extension of the LOTOS formal description technique. Computer Communications 23(12) (2000) 1104–1123. https://10.1016/S0140-3664(99)00240-6.
20. Santos, J. dos, Braga, C., Muchaluat-Saade, D. C. (2013). , An Executable Semantics for a Multimedia Authoring Language. , in *Formal Methods: Foundations and Applications. SBMF 2013. Lecture Notes in Computer Science*, vol. 8195, pp. 67–82., (2013). , https://10.1007/978-3-642-41071-0.
21. Rutledge, L. SMIL 2.0: XML for web multimedia. IEEE Internet Computing 5(5) (2001) 78–84. https://10.1109/4236.957898.
22. Carlsson, M. Monads for incremental computing. ACM SIGPLAN Notices 37(9) (2002) 26–35. https://10.1145/583852.581482.