Novel Framework for Translation Algorithms in Multimedia Authoring Tools

Marvin Chandra Wijaya, Zulisman Maksom, Muhammad Haziq Lim Abdullah

Abstract— **Multimedia authoring tools can transform a multimedia presentation by converting its temporal and spatial layout into a multimedia document. Multimedia documents widely used are a multimedia programming language called Synchronized Multimedia Integration Language (SMIL). The problem is the processing time of multimedia authoring, which takes time when creating multimedia documents. The process in multimedia authoring tools consists of editing, verification, modeling, and translation. This study proposes a novel framework by combining incrementality in the verification process and creating a new translation algorithm. Incrementality has a function to perform pre-verification in a short time on multimedia models, which can reduce the time required for verification using time computation and spatio-temporal verification. Experiments are conducted by inserting various media objects to determine a multimedia document's accuracy and processing time. The proposed framework successfully detects all given spatial and temporal overlaps. Based on experimental results, the time required to process verification using the proposed framework has reduced processing time by 77.81%. The framework studied in this research is to modify the verification process and increase the efficiency of the translation algorithm. Reducing processing time will make it easier for designers to create a multimedia presentation.**

Index Terms— **multimedia authoring tools, incrementality, verification, translation Algorithm, SMIL**

I. Introduction

Currently, multimedia is one way of delivering information that is widely used from presenters to audiences. The use of multimedia has increased because of the many features that can be used in multimedia content. A Multimedia authoring tool is needed to create multimedia content, producing multimedia documents. Multimedia authoring tool should help designers create multimedia documents efficiently, quickly, and well [1]. A multimedia document is an inventory of multimedia information that contains an application model and multimedia objects [2].

Manuscript received November 19, 2021; revised May 21, 2022. This research was supported by Maranatha Christian University, Indonesia and Universiti Teknikal Malaysia Melaka, Malaysia.

Marvin Chandra Wijaya is a Ph.D candidate of Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Malaysia. He is also a lecturer of Maranatha Christian University, Indonesia (corresponding author e-mail: p031910033@student.utem.edu.my).

Zulisman Maksom is a senior lecturer of Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Malaysia. (e-mail: zulisman@utem.edu.my).

Muhammad Haziq Lim Abudllah is a senior lecturer of Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Malaysia. (email: haziq@utem.edu.my)

A multimedia programming language represents the multimedia document. Multimedia programming languages used in multimedia documents include Nested Context Language (NCL), Hypertext Markup Language 5 (HTML 5), and Synchronized Multimedia Integration Language (SMIL) [3], [4]. SMIL and NCL are part of the Extensible Markup Language (called XML) for data exchange needs using Data-Type-Definitions (DTD) as a model in their multimedia documents [5]. Multimedia programming languages can also serve to write interactive audiovisual presentations by designers. These programming languages define layout, timing, animation, visual transitions, interactive methods, shifts, and more [6], [7]. Multimedia documents can be retrieved from several servers or cloud network facilities using the internet network [8].

The distribution of multimedia documents needs to be safeguarded and protected to protect digital privacy. Algorithms and routing protocols are required to manage the distribution of multimedia documents according to demand or using online web-based distribution [9]–[11]. The distribution of multimedia presentations can use several channels, such as peer-to-peer methods (local computer networks, Wi-Fi direct) and broadcast methods (Wi-Fi, 3G, 4G, LTE, or 5G systems). The integrity of the data sent over the network must be verified beforehand [12]. Multimedia applications distributed over the internet have been widely applied in various fields such as games, mobile-learning, multimedia sensors, video streaming, e-health, e-commerce, e-learning, military, tourism, movie, paper works, information, problem solving, e-travel, internet radio, and others [13]–[17].

Although the use of multimedia programming languages is intended to facilitate efforts to write multimedia presentations, there is a possibility that the spatio-temporal layout generated by multimedia authoring is not under the expectation of the author. This error is caused by errors in the use of constructs in the multimedia programming code used by the author. This error is called a mismatch between the results expected by the designer and the multimedia presentation produced by the multimedia authoring tool. Therefore, verification of the multimedia authoring tools is necessary to make sure that the results of the multimedia presentation generated are as expected by the author.

There are several steps to using a multimedia authoring tool: media object inventory entry, editing, verification, modeling, and translation[18]. While entering the media object inventory, the designer must enter the media object data into the multimedia authoring tools. The author enters data in the form of temporal and spatial layout during the editing process [19], [20]. Multimedia elements consist of media objects, temporal layout, spatial layout and user interface/interaction, and visual style, as shown in Fig 1.

These elements are part of application logic, such as multimedia authoring tools.

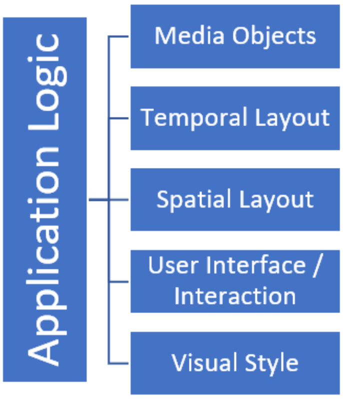

Fig. 1. Elements multimedia

The following process is to verify the designer's input to check if conflicts occur. The verification process usually takes a long time, especially if the designer makes a lot of changes, both major and minor changes. The modeling and translation processes also require a long time, as in the verification process. Overall, these processes must have an efficient algorithm. The modeling and translation processes also require a long processing time, as with the verification process. There are several multimedia models for kernel processing in multimedia authoring tools, such as simple interactive multimedia models (SIMM), hoare logic, and petri nets. [21].

These processes depend on the size of the multimedia document. The more media objects in the multimedia document, the longer it will take. The comparison between the number of media objects and the time required is shown in Fig 2. The need for efficient frameworks and algorithms is essential and needed by multimedia authoring tools.

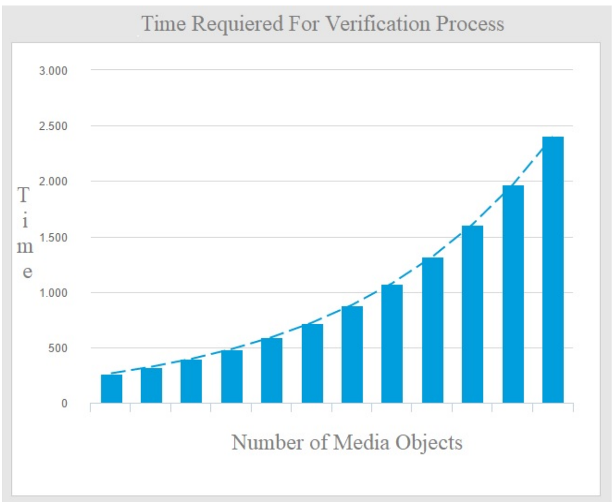

Fig. 2. Comparison of the time required with the number of multimedia objects

II. Related Works

Figure 3 shows several steps in the process of a multimedia authoring tool, as discussed in the introduction section.

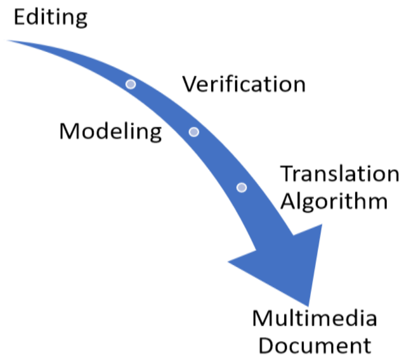

Fig. 3. Multimedia Authoring processes

A. *Editing Process*

In multimedia authoring tools, editing facilities are given to designers to create multimedia presentations. There are two attributes of measuring this editing process: ease of use and expression [22]. Editing attributes in multimedia authoring tools include temporal view, spatial view, media inventory, message zone, and text editor, as shown in Fig 4. An initial factor of user convenience on display is the "What You See Is What You Get" factor. This initial factor was investigated in 2003 by Yang and Yang [20]. The editor view is divided into layout, timeline, preview, and attribute views.

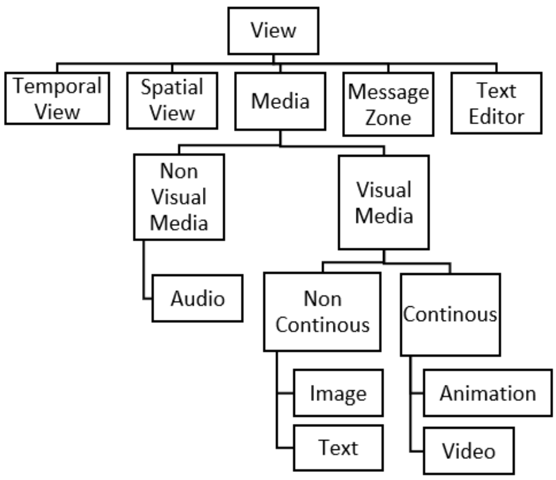

Fig. 4. Hierarchy of multimedia user interface

Spatio-temporal is a representation of multimedia presentations that are widely used for various purposes [23]. In 2004, Mee Young Sung implemented a spatial and temporal layout model into a 3-dimensional model called the spatio-temporal [24]. The designer can also edit the model in a temporal relation network (TRN). Three-dimensional spatial-temporal is a three-dimensional model that shows the spatial layout and temporal layout in one graph, as shown in Figure 5. Each graph axes represent temporal information (1 axis) and spatial information (2 axes). Each media object in the form of visual (image, video, animation) is represented in a rectangular box, and a cylinder represents every non-visual object media (audio).

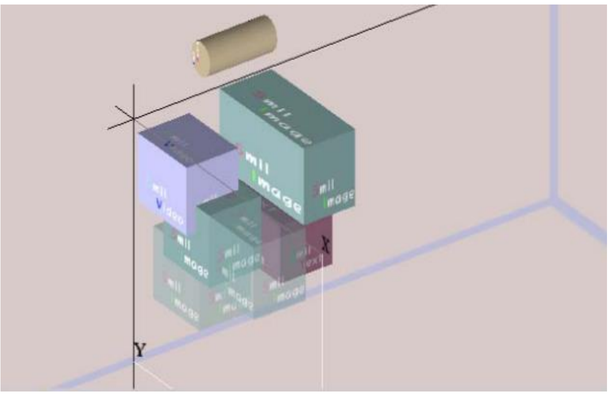

Fig. 5. Temporal relation network

Another factor in the editing process is expressivity which consists of document structure, characteristic specifications, media objects, composition domain, flexibility in temporal layout, spatial layout, complex relationships, and adaptation in multimedia presentation [25]. Even simple cut, copy, and paste facilities have helped in the editing process and are mandatory features of the multimedia authoring tools user interface[26].

B. Verification Process

The process of modeling and verification are two processes that cannot be separated from other processes. Both processes are essential processes in a multimedia authoring tool. Multimedia presentation needs to be appropriately verified to reduce errors in the following procedure. Spatial layout and temporal layout need to be verified to check conflicts in the media objects. [27].

The time calculation strategy checks the consistency of time on the overall time duration of the multimedia presentation. Whenever the temporal layout of a multimedia presentation is changed, the verification process is carried out at the low level first. After verification process has been carried out at the low level, it will proceed to a higher level. The verification process will be stopped at each level if errors are found and then provide error warnings to the designer via the message area [28]. Time computation is needed to detect temporal overlap, as shown in Fig 6.

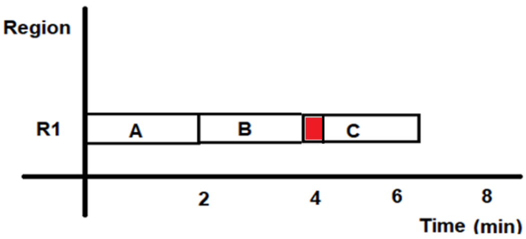

Fig. 6. Temporal conflict

Errors can occur due to the overlap of areas where media objects are located, called spatial conflict [29]. Spatial conflict can occur in visual media objects due to one, two, or more overlapping areas that place media objects simultaneously. Overlapping between two areas can occur because the two areas cover all other areas (Entire spatial conflict) or only cover part of the area (partial spatial conflict), as shown in Fig 7.

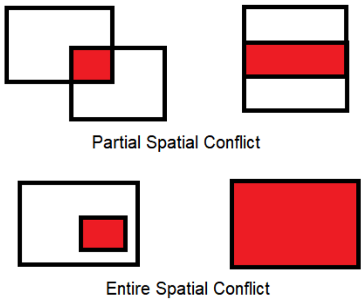

Fig. 7. Spatial conflict

There are two approaches to the verification process that can be applied: before and after the presentation begins. The verification approach before the presentation media starts is not ideal compared to the presentation media that has already started. Dynamic spatio-temporal verification is executed from the beginning of the presentation by monitoring the object's media area.[30].

C. Modeling Process

Multimedia modeling serves to represent a multimedia presentation in various ways. Multimedia modeling serves to represent multimedia presentations in various ways. Several models for multimedia modeling include hoare logic, petri net with several variations, Simple Interactive Multimedia Model (SIMM), and language of temporal ordering specifications.

Petri Net is multimedia modeling using a graphical representation. Petri Net consists of several components: place, transition, arc, and token. The token is located in a place and will move from one place to another—a simple Petri Net graph as shown in Fig 8.

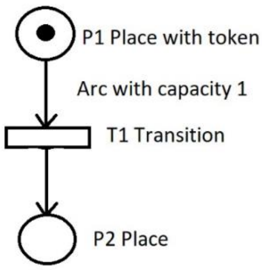

Fig. 8. Simple petri net graph

D. Translation Process

The translation process serves to convert a multimedia model into a multimedia document. In 2003 Yang and Yang conducted a study on a translation algorithm called Scan2SMIL to translate a multimedia model into a multimedia document [20].

The Scan2SMIL algorithm is as follows:

Step 1. Create a parallel element.
Step 2. Sort all objects and name the table with table object.
Step 3. While the object still exists in the table object, do {
Step 4. Select the smallest object in the object table.
Step 5. Add the object to the set S.
Step 6. Find the nearest object that is not connected to the object in the set S.
Step 7. Add found Objects

Step 8. Add unconnected objects until all objects have been scanned.
Step 9. Remove the object on S from the table object.
Step 10. Objects in S are created sequential elements as children of parallel elements that are created in step 1.}

E. Multimedia Document

Multimedia documents are represented in the form of a multimedia programming language. Multimedia programming languages used in multimedia documents include Nested Context Language (NCL), Hypertext Markup Language 5 (HTML 5), and Synchronized Multimedia Integration Language.

The World Wide Web Consortium also recommends the SMIL language, a markup-like language, and a subset of web programming (HTML). The synchronized multimedia integration language is a part of extended markup language (XML). SMIL can control multimedia presentations. The structure of the SMIL programming language consists of a head and a body, as shown in Fig 9.

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
  ... information of meta, layout, title,
      and transistion effect ...
  </head>
  <body>
  ... media, group, linking, animation...
  </body>
</smil>
```

Fig. 9. SMIL structure

There are many features that SMIL can control. SMIL can also be used to create MMS content, a multimedia-based messaging service on cellular technology.
There are many advantages to using SMIL to develop multimedia presentations:

- Multiple media can be placed on different servers or cloud servers.
- Setting the layout of the media object.
- Set the time and synchronization of media objects.
- Create interactive multimedia.
- Creation of links to web pages.
- Media object transition settings.
- Different presentation settings for other users.

Another widely-used programming language is nested context language (NCL). NCL and SMIL are XML-based declarative languages. NCL has a modular form for incorporating modules into language profiles. NCL has multimedia presentation capabilities by creating application content on various devices according to the temporal layout for media objects to be played. NCL separates the structure and content of media objects in multimedia presentations. The commands in the NCL will arrange each media object according to its spatial layout and temporal layout.

The structure of the NCL programming language consists of a head and a body, as shown in Fig 10.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ncl>
<head>
      <regionBase>
        . . .
      </regionBase>
      <descriptorBase>
        . . .
      </descriptorBase>
      <connectorBase>
        . . .
      </connectorBase>
</head>
<body>
   . . .
</body>
</ncl>
```

Fig. 10. NCL structure

III. METHOD

As discussed in the previous section, the standard processes of Multimedia Authoring consist of the editing process, verification process, modeling process, and translation process to produce multimedia documents. This study proposes a novel framework to run these processes more efficiently, as shown in Fig 11. In order to improve the efficiency of the multimedia authoring tools process, it is focused on several things that become bottlenecks in certain processes. In accordance with the literature study results, it was found that two things are of concern in improving efficiency. The first is the time-consuming verification process, which has an exponential relationship with many objects being processed. The second is the translation algorithm to translate the multimedia model into a multimedia document.

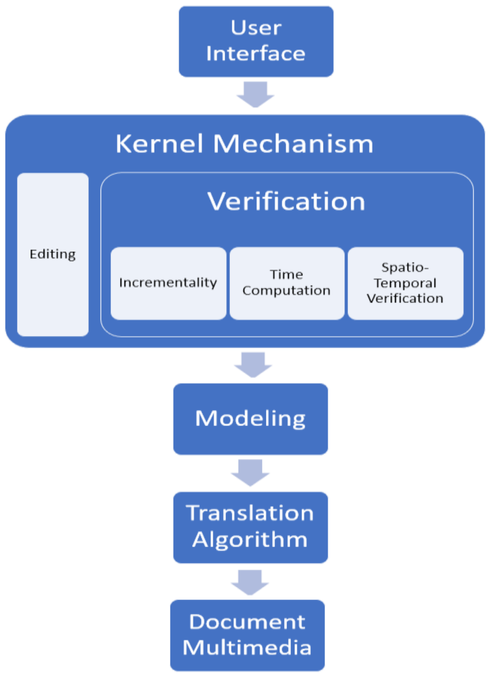

Fig. 11. A Novel Framework of Multimedia Authoring Processes

In this study, a verification system with three stages is proposed. The first stage of verification is an incrementality process that eliminates initial errors made by designers when designing a multimedia presentation. The second stage of verification is time computation, which verifies errors if there is time overlap for the same area. The third stage of verification is spatio-temporal which serves to verify errors in case of time overlapping for different areas that overlap each other.

A. Incrementality Verification

A verification system with the incremental method can detect errors in a multimedia authoring tool more quickly than a system without using the incremental method [28], [31]. The system with the incremental method is able to process verification with less computation time than the previous system.

The algorithm for verification by the incremental method consists of ten steps:

Step 1. While object {
Step 2. Input object_media.
Step 3. Categorization of object_media as visual media or non-visual media.
Step 4. If the media is not found then notify the author.
Step 5. If the visual media is not found then notify the author.
Step 6. If the non-visual media is not found, then notify the user.
Step 7. Input object_media begin, end and duration.
Step 8. If Object_media begin, end and duration is not consistent then notify the author.
Step 9. For visual object_media, input region.
Step 10. If the region is not found, then notify the user. }

B. Time Computation Verification

After the designer's errors have been detected and corrected, the following process is time computation verification. The time computation verification will calculate the overall time duration of the multimedia model used. Calculation and verification are carried out in stages from the local level first [32].

An example of a verification process using a time computing algorithm is a time change given to a media object. If the media object video changes from 500 seconds to 600 seconds, the verification process will first be carried out at the lowest level. In step 1, the change in duration from 500 seconds to 600 seconds will be calculated, or parallel duration will be calculated from 500 seconds to 600 seconds. If the error has been fixed or there are no errors at that level, then the verification process is continued to the next level in the body from 1000 seconds to 1100 seconds

C. Spatio-temporal Verification

Spatial conflicts in multimedia documents will not cause major errors but make multimedia presentations imperfect. Spatial conflicts occur because of overlapping media placement in several areas in a root layout. Non-visual objects, such as music and audio, overlap in the temporal layout. Visual objects (video, images, text, and animation) overlap in the spatial layout as well as the temporal layout. Two areas will be considered as overlapping areas if at the same time, two media objects are played; this happens if the following Boolean equations are met equation (1).

$$((Left2>Left1)\ \&\&\ (Left2<Left1+Width1))\ \|\ ((Left2+Width2>Left1)\ \&\&\ (Left2+Width2<Left1+Width1))\ \&\&\ ((Top2>Top1)\ \&\&\ (Top2<Top1+Height1))\ \|\ ((Top2+Height2>Top1)\ \&\&\ (Top2+Height2<Top1+Height1)) \qquad (1)$$

Where:
Top1 = Top position of media object 1
Left1 = Left top position of media object 1
Top2 = Top position of media object 2
Left2 = Left position of media object 2
Height1 = Height of media object 1
Width1 = Width of media object 1
Height2 = Height of media object 2
Width2 = Width of media object 2

D. Linear Translation Algorithm

The translation algorithm is the last process in the kernel mechanism that produces a multimedia document from a multimedia presentation designed by the designer. The translation algorithm must consider spatial and temporal information from all media objects. Spatial information is related to the layout of the media, which is called a region. The period of media presented is called temporal information, which contains data about when a media starts and ends.

In this study, a novel algorithm is made called the linear translation algorithm. In this algorithm, each object is classified according to the media type, and each object for each media type forms a child element and a root <par> element in the SMIL Code. A combined play period is created by observing the conditions of spatial-temporal conflict on visual objects (videos, pictures) and the position of the same area. For each media object with the same regional position, a <seq> tag is created.

The Linear Translation Algorithm is as follows:

Step 1. Create root <par> element on top
Step 2. Classify object by medium type (visual object or non-visual object)
Step 3. While object {
Step 4. For visual object {
Step 5. Classify object by region/layout position
Step 6. Object of each display position create <seq> element
Step 7. All <seq> elements of different display position form a <par> element }
Step 8. For non-visual object {
Step 9. Object create <seq> element } }

E. Verification Strategy

The spatio-temporal verification in this study uses shifting two or more media objects. The shift will have at least two media objects that shift until they collide or overlap. Figure 12 shows an example of a shift in object media A that shifts continuously until it collides with object media B.

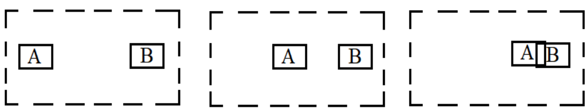

Fig. 12. Media Object shifting

The experiment contains many shift steps for each media object whose number of media objects has been determined. The number of media objects that collide or overlap will be selected randomly. Media objects will overlap with other media objects in the same region or area in order to verify the time computation process. This condition is used as experimental material to see the success of the verification process. In addition, there are overlapping media objects in different areas to verify spatio-temporal verification.

IV. RESULT AND DISCUSSION

The experiment conducted in this study is to test the framework by providing temporal layout data consisting of 2 regions (R1 and R2). There are two media objects in region 1 (R1) (Img1.jpg and Vid2.mov). In region 2 (R2), two objects (Vid1.mov and Img2.jpg) are shown in Fig 13.

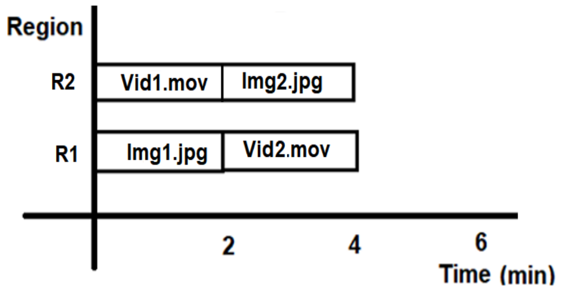

Fig. 13. Experiment scenario 1

After the experimental scenario 1 was entered into the multimedia authoring tools, it turned out that the multimedia document was successfully obtained correctly as follows:

```
<smil>
<head>
<layout>
<root-layout height="660 width="660" " />
<region id="R1" height="355" width="305" left="15"
top="18" />
<region id="R2" width="355" height="355" left="15"
top="360" />
</layout>
</head>
<body>
<par>
<seq>
    <video  src="Vid1.mov" start=”0” dur="120"
region="R2" />
    <video  src="Img2.jpg" start=”1200” dur="120"
region="R2" />
</seq>
<seq>
    <image  src="img1.jpg" region="R1" start=”0”
dur="120"/>
    <image  src=”vid2.mov" region="R1" start=”120”
dur="120"/>
</seq>
</par>
</body>
</smil>
```

An experiment with other scenarios conducted in this study is to test the framework by providing temporal layout data consisting of 2 regions (R1 and R2). In region 1 (R1), there are two media objects (Img1.jpg and Img2.jpg). In region 2 (R2), there are two objects (Vid1.mov and Vid2.mov) and one non-visual media object (Song1.mp3), as shown in Fig 13.

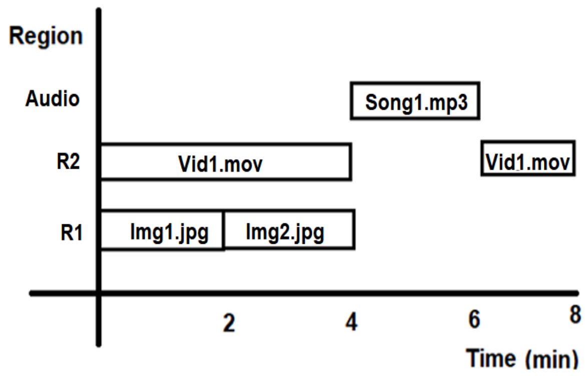

Fig. 14. Experiment scenario 2

After the experimental scenario 2 was entered into the multimedia authoring tools, it turned out that the multimedia document was successfully obtained correctly as follows:

```
<smil>
<head>
<layout>
<root-layout height="1025" width="1055" />
<region id="R1" height="360" width="460" left="50"
top="50" />
<region id="R2" width="460" height="360"  left="50"
top="550" />
</layout>
</head>
<body>
<par>
<audio src=”Song1.mp3” start=”240”>
<seq>
    <video src="vid1.mov" region="R2" start=”0”
dur="240"/>
    <video src="vid2.mov" region="R2" start=”360”
dur="120"/>
</seq>
<seq>
    <image src="img1.jpg" region="R1" start=”0”
dur="120"/>
    <image src="img2.jpg" region="R1" start=”120”
dur="120"/>
</seq>
</par>
</body>
</smil>
```

Based on the two test scenarios, multimedia documents in the SMIL programming language have been produced. The multimedia document is in accordance with the two scenarios given.

Another experiment was conducted to determine the success rate of spatial-temporal verification and time computation by giving several overlaps in the area and on the timeline. Overlapping on the area is given to show the success of spatial-temporal verification, and overlapping on the timeline is given to show the success of time computation. In this experiment, the number of overlaps is given randomly to determine the success of the designed algorithm. The results of the algorithm are checked how much overlapping has been detected.

Table 1 and Fig 15 show that overlapping detection using spatial-temporal verification and time computation has succeeded in detecting all (100%) of the given overlap. Several overlaps ranging from 15 to 51 overlapping media were verified using spatiotemporal verification and time computation. The number of overlappings detected by both verification methods is the same as the number of overlaps given.

TABLE I
EXPERIMENT RESULTS OF OVERLAPPING DETECTION

Number of Overlappings	Number of detections		Succesful detection rate
	Spatial-temporal verification	Time computation	
15	11	4	100%
18	14	4	100%
22	15	7	100%
28	20	8	100%
33	24	9	100%
37	25	12	100%
39	26	13	100%
42	29	13	100%
47	32	15	100%
51	35	16	100%

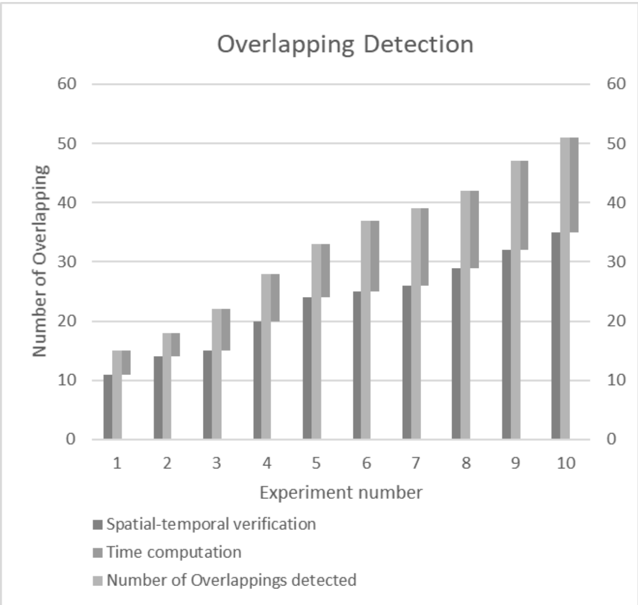

Fig. 15. Overlapping detection

The framework testing strategy will be carried out by comparing the framework proposed in this study with the existing framework, as shown in Fig 16. The difference is in the existing frame without using an incremental verification process and a standard translation algorithm. Experiments are carried out by giving the number of different media objects (10 to 100) and then measuring the time needed to produce a multimedia document, as in Table 2 and Fig 17.

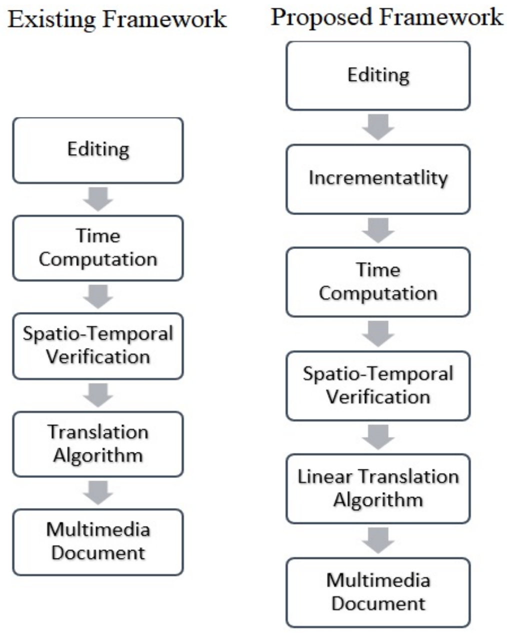

Fig. 16. Testing Strategy

TABLE II
EXPERIMENT RESULTS OF THE PROPOSED FRAMEWORK

Number of Objects	Time Required (Existing Framework)	Time Required (Proposed Framework)	Time Reduction
10	207 ms	45 ms	78.26%
20	307 ms	68 ms	77.85%
30	452 ms	101 ms	77.65%
40	880 ms	180 ms	79.55%
50	1127 ms	270 ms	76.04%
60	1727 ms	399 ms	76.90%
70	2576 ms	580 ms	77.48%
80	3799 ms	874 ms	76.99%
90	5653 ms	1270 ms	77.53%
100	8071 ms	1630 ms	79.81%

Processing time measurements are carried out for each experiment using the existing framework and proposed framework. Based on the experimental results, the processing time on the multimedia authoring tool was reduced by an average of 77.81%. Reducing processing time will make it easier for designers to create a multimedia presentation. The experimental results have limitations with the increasing number of media objects, the longer the processing time.

V. CONCLUSION

This study proposed two new improvements in the multimedia authoring tools processes. The first process is to create a new verification process by combining the multimedia model's incrementality process. The second process is to create a linear translation algorithm that can convert multimedia representations into multimedia documents. The proposed framework successfully detects all given spatial and temporal overlaps. Based on experimental results, the translation process successfully produced multimedia documents correctly and reduced processing time by 77.81%. Reducing the size of multimedia documents to make them easier to distribute can be studied in the future.

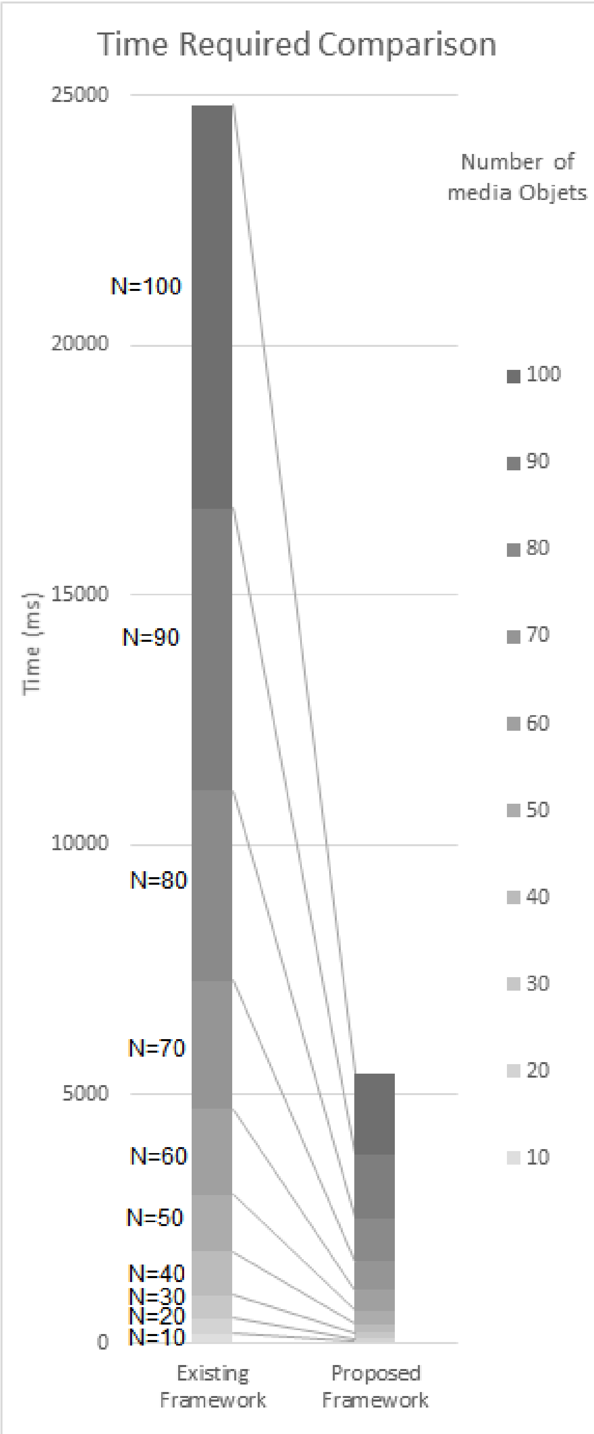

Fig. 17. Time Comparison between Existing Framework and Proposed Framework

REFERENCES

[1] D. Picinin, J. M. Farines, C. A. S. Santos, and C. Koliver, "A design-oriented method to build correct hypermedia documents," *Multimedia Tools and Applications*, vol. 77, no. 16, pp. 21003–21032, 2018.

[2] Y. Belhadad, A. Refoufi, and P. Roose, "Spatial reasoning about multimedia document for a profile based adaptation: Combining distances, directions and topologies," *Multimedia Tools and Applications*, vol. 77, no. 23, pp. 30437–30474, 2018.

[3] R. L. Guimarães, P. Cesar, and D. C. A. Bulterman, "Creating and sharing personalized time-based annotations of videos on the web," *DocEng2010 - Proceedings of the 2010 ACM Symposium on Document Engineering*, pp. 27–36, 2010.

[4] M. Jourdan, "A formal semantics of SMIL: A web standard to describe multimedia documents," *Computer Standards and Interfaces*, vol. 23, no. 5, pp. 439–455, 2001.

[5] S. C. Haw and E. Soong, "Performance evaluation on structural mapping choices for data-centric XML documents," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 3, pp. 1539–1550, 2020.

[6] P. Hoschka, "The application/smil and application/smil+xml Media Types," *RFC*, vol. 4536, pp. 1–8, 2006.

[7] M. U. Mwinyi, J. M. Alja'am, and S. A. ElSeoud, "A concept analysis scheme of simple stories for learning resources through extraction of domain-based multimedia elements," *International Journal of Interactive Mobile Technologies*, vol. 11, no. 2, pp. 53–65, 2017.

[8] S. Banerjee, C. Chakraborty, and S. Paul, "Programming Paradigm and the Internet of Things," in *Handbook of IoT and Big Data*, 2019, pp. 145–164.

[9] A. Maheshwari, S. Agrawal, H. Venkataraman, and G. M. Muntean, "PRIoritized multimedia adaptation scheme over two-hop heterogeneous wireless networks (PRiMA)," *Engineering Letters*, vol. 18, no. 2, pp. 119–134, 2010.

[10] Y. W. Syaifudin *et al.*, "A web-based online platform of distribution, collection, and validation for assignments in android programming learning assistance system," *Engineering Letters*, vol. 29, no. 3, pp. 1178–1193, 2021.

[11] P. Maniriho and T. Ahmad, "Enhancing the capability of data hiding method based on reduced difference expansion," *Engineering Letters*, vol. 26, no. 1, pp. 45–55, 2018.

[12] A. E. Mae, W. K. Kwee, P. Y. Han, and L. S. Hoe, "Resource Allocation for Real-time Multimedia Applications in LTE's Two-level Scheduling Framework," *IAENG International Journal of Computer Science*, vol. 43, no. 4, pp. 427–437, 2016.

[13] E. Hendarwati, L. Nurlaela, and B. S. Bachri, "Collaborative Problem-Solving Based on Mobile Multimedia," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 13, pp. 16–27, 2021.

[14] S. Suryanti, W. Widodo, and Y. Yermiandhoko, "Gadget-Based Interactive Multimedia on Socio-Scientific Issues to Improve Elementary Students' Scientific Literacy," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 1, pp. 56–69, 2021.

[15] N. Edan and E. Y. Abdullah, "Design and implementation of a novel secured and wide WebRTC signalling mechanism for multimedia over internet," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 5, pp. 5430–5435, 2020.

[16] Y. N. Prajapati and M. K. Srivastava, "Novel algorithms for protective digital privacy," *IAES International Journal of Robotics and Automation*, vol. 8, no. 3, pp. 184–188, 2019.

[17] Y. J. I. A. Yghoubi, W. L. Pang, S. K. Wong, and K. Y. Chan, "Performance evaluation of video streaming on LTE with coexistence of Wi-Fi signal," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 3, pp. 890–897, 2019.

[18] B. Huang, S. Yi, and W. T. Chan, "Spatio-temporal information integration in XML," *Future Generation Computer Systems*, vol. 20, no. 7, pp. 1157–1170, 2004.

[19] C. Yang, C.-K. Chu, and Y.-C. Wang, "Extension of Timeline-based Editing for Non-deterministic," *Journal Of Information Science and Engineering*, vol. 24, no. 5, pp. 1377–1395, 2008.

[20] C. C. Yang and Y. Z. Yang, "SMILAuthor: An Authoring System for SMIL-Based Multimedia Presentations," *Multimedia Tools and Applications*, vol. 21, no. 3, pp. 243–260, 2003.

[21] M. C. Wijaya, Z. Maksom, M. Haziq, and L. Abdullah, "A Brief Review : Multimedia Authoring Modeling," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 13, no. 1, pp. 39–48, 2022.

[22] M. C. Wijaya, Z. Maksom, and M. H. L. Abdullah, "A Brief of Review : Multimedia Authoring Tool Attributes," *Ingénierie des Systèmes d'Information*, vol. 26, no. 1, pp. 1–11, 2021.

[23] C. Chen, Y. Wang, K. Yi, T. Wang, and H. Xiang, "Semantic analysis of action with spatio-temporal features based on object detection," *Engineering Letters*, vol. 28, no. 2, pp. 616–623, 2020.

[24] M. Y. Sung and D. H. Lee, "A collaborative authoring system for multimedia presentation," *IEEE International Conference on Communications*, vol. 3, no. c, pp. 1396–1400, 2004.

[25] M. J. Antonacci, D. C. Muchaluat-Saade, R. F. Rodrigues, and L. F. G. Soares, "Improving the expressiveness of XML-based hypermedia authoring languages," in *Multimedia Modeling: Modeling Multimedia Information and Systems, MMM 2000*, 2020, pp. 71–88.

[26] M. Jourdan, C. Roisin, L. Tardif, and L. Villard, "Authoring SMIL documents by direct manipulations during presentation,"

World Wide Web, vol. 2, no. 4, pp. 179–190, 1999.

[27] S. Bouyakoub and A. Belkhir, “A Spatio-Temporal Authoring Tool for Multimedia SMIL Documents,” *International Journal of Multimedia Technologies*, vol. 2, no. 3, pp. 83–88, 2012.

[28] D. P. Júnior, J. M. Farines, and C. Koliver, “An approach to verify live NCL applications,” in *WebMedia’12 - Proceedings of the 2012 Brazilian Symposium on Multimedia and the Web*, 2012, pp. 223–231.

[29] L. Rutledge, “SMIL 2.0: XML for web multimedia,” *IEEE Internet Computing*, vol. 5, no. 5, pp. 78–84, 2001.

[30] J. A. F. Dos Santos, C. Braga, C. Débora, C. Roisin, and N. Layaïda, “Spatio-temporal validation of multimedia documents,” in *Proceedings of the 2015 ACM Symposium on Document Engineering*, 2015, no. September, pp. 133–142.

[31] M. Carlsson, “Monads for incremental computing,” *ACM SIGPLAN Notices*, vol. 37, no. 9, pp. 26–35, 2002.

[32] M. C. Wijaya, Z. Maksom, and M. H. L. Abdullah, “Two Verification Phases in Multimedia Authoring Modeling,” *Journal of information and communication convergence engineering*, vol. 19, no. 1, pp. 42–47, 2021.

Marvin Chandra Wijaya is a researcher and senior lecturer at Maranatha Christian University, Indonesia. Areas of interest include multimedia, artificial intelligence, image processing, and microcontroller.

Zulisman Maksom is a researcher and senior lecturer at Universiti Teknikal Malaysia Melaka, Malaysia. Areas of interest include instructional design, human-computer interaction, pedagogy technology, and graphics design.

Muhammad Haziq Lim Abdullah is a researcher and senior lecturer at Universiti Teknikal Malaysia Melaka, Malaysia. Areas of interest include interaction design (people with special needs), serious game, human-computer interaction, augmented reality, assistive technology, and co-design.