Abstract: Several requirements visualization techniques have been developed together with tools that support the techniques. This paper focuses on visualization techniques developed for requirements representation. These techniques are used to transform the written and verbal requirements into visual representations. Most of the existing techniques use graphics for visual representation. This paper reviews three requirements graphical visualization techniques; which are the 3D visualization technique, UML models, and behavioural requirements specification approach. These three are later compared using four selected attributes which are ease of use, validation support, comprehensiveness, and ease of conversion to implementation codes.

Key-Words: Requirements Visualization Technique, Graphical Visualization Techniques, Requirements Engineering

1 Introduction
Research in requirements engineering (RE) still needs a lot of improvements. Many new critical RE research questions have been raised as the technology demands have also been increasing tremendously [1]. Requirements visualization is not new [2], a lot of work has been done on it and many techniques and tools have been developed. The domain keeps on evolving to be more stable.

RE is a very important activity in the software engineering practice. However, it is not easy to define good requirements. RE can be difficult as it involves a lot of iterations, players, analysis, and complicated verifications [1]. There are seven distinct functions of RE as stated by Pressman. These are inception, elicitation, elaboration, negotiation, specification, validation, and management [3].

Requirements visualization is defined as a method that comprehends requirements represented by using diagrams or graphics [4]. Its goal is to arouse consciousness and insights while transforming requirements to make developers assimilate easier [5]. According to Marchese et al., visualization involves the process of gathering, processing, pictorial rendering, analyzing and interpreting data or requirements [5]. Requirements visualization helps requirements validation.

Numerous techniques have been produced by RE research, mainly to improve the quality of the defined requirements. Most research focus on visualizing functional requirements. This paper reviews three existing functional requirements graphical visualization techniques.

The structure of this paper is as follows. In the next section, we summarize the three techniques. In the following section, we compare the techniques through four selected significant attributes. Lastly, we give our conclusion.

2 Requirements Visualization Techniques
Requirements visualization techniques help translate textual or verbally defined requirements into visual representations which are much easier understood. Three types of visualization approaches have been chosen. They are the 3D requirements visualization technique, UML visualization technique, and behavioural requirements visualization technique. All these techniques are meant for functional requirements visualization.

2.1 3D Visualization Technique
The idea of 3D visualization technique came about in the 1990s and was discussed in Robertson et al. 3D visualization involves bigger physical space resulting in higher density of information compared to 2D [6]. Teyseyre used the approach in his research and developed a tool named ReqViz3D [4]. The tool helps validation of requirements reducing the communication gap between the
customers and the developers. The aim of developing ReqViz3D was to improve the efficiency of the requirements validation process. Requirements are elicited and then documented using language Z. The requirements are then translated and visualised using 3D animation which facilitates user validation. The process is depicted in Fig. 1.

Fig. 1: Global System View of ReqViz3D

Teyseyre developed ReqViz3D using JAVA programming language. The requirements are defined using the language Z. These are then translated by a translation procedure into the PROLOG programming language. The tool then visualizes and animates the specification concepts in 3D (see Fig. 2).

The visual representations are similar to the real world. The author claimed that users only need to define the requirements and leave the other tasks to the tool. However, the construction process of visual representations is difficult and time consuming.

Fig. 2: An example of Lift System Visualization using ReqViz3D

2.2 UML Visualization Technique

Unified Modelling Language (UML) models have been used widely in assisting requirements visualization. It is one of the most widely used approaches by requirements engineers and in academia. This is due to the high understandability and learnability of UML diagrams and notations which results in development of many open source UML CASE tools.

REVU (Requirements Visualization of UML) [7] is an approach that has the following steps: create witness scenario properties in natural language, generate witness scenarios from witness traces, and represent scenario into UML models.

REVU approach is an integration of existing approaches: SPIDER [8], Hydra [9] and the Theseus tool [10]. SPIDER (Specification Pattern Instantiation and Derivation EnviRonment), which is supported by a tool suite, is used in REVU to specify properties in UML models from a natural language. Hydra then automates the process of mapping UML models into a formal specification language. Theseus, a UML visualization tool, is used for the visualization of the witness scenarios.

An example of an output from the visualization approach is shown below (see Fig. 3).

Fig. 3: An example of visual representation of requirements using sequence diagram

2.3 Behavioural Visualization Technique

Another technique of requirements visualization focuses on the behaviour of the requirements [11]. The research was aimed to help visualize the requirements of multimedia applications including interactions. The elements in multimedia applications are classified into two categories: transitions due to event (the transition trigger - automatic, user, or clock) and transitions due to task (the system’s action - animate, image, sound, text or video).

The notations used in the approach [12] and the toolkit they developed were adapted from Petri Net.

Petri Net notations used for RNSMA:

- Place which represent the state of the system.
- Transition which causes changes to the state of the system.
- Connects a place to a transition or vice-versa.

Khairuddin et al. extended the Petri Net notations for multimedia application elements through their Requirements Notation for Specifying
Multimedia Applications (RNSMA) (see Fig. 5 and Fig. 6). The notations are capable of modelling sequential, parallel, conditional and iterative statements.

![Notations](image.png)

Fig. 5: Notations for Transitions Due to Events Category

![Notations](image.png)

Fig. 6: Notations for Transitions Due to Tasks Category

3 Discussion

All three techniques discussed in this paper use graphics to visualize the requirements. It is the most used technique type for visualization. All of the three approaches discussed developed tools to support their techniques of visualizing functional requirements. In this section, a comparative analysis is done based on four significant attributes. The differences between the three techniques including their advantages and disadvantages are discussed.

The first attribute is ease of use. This is a very important attribute. Developers want to choose techniques and tools that are easy for them to use and not too complex to understand. This attribute covers comprehensibility. This will ensure every developer is capable of understanding how to use the technique without any difficulty.

The second attribute chosen is validation support. It should be easy for the customers to validate requirements. This covers ease of understanding the visualizations. Validation will ensure the requirements meet users’ needs.

The third aspect to be considered is comprehensiveness. This attribute covers features of the technique that allow them to represent different aspects of requirements such as data flow, control flow, relationships, sequence and states.

The fourth comparison attribute is ease of conversion to implementation codes. Requirement specification that uses formal method would make the conversion much easier. Clear and simple diagrams will also help simplify conversion.

The comparison of the three techniques based on the four attributes is given in Table 1. Each of the techniques reviewed has its strengths and weaknesses.

3D visualization approach using language Z requires a high understanding of formal methods. However, formal methods allow for ease of conversion. Another of its advantages is that 3D visualization results are very close to the real world. This makes it highly understandable and very easy to be validated. However, simulation construction is difficult and time consuming [4].

UML-based approach is dependent on the object oriented concept which is quite unfamiliar to those who were trained to work using procedural languages. However, this approach has gained popularity especially through young developers who are exposed to the object-oriented techniques and tools. There are also many versions of CASE tools, reference books, and web pages to assist in the design of the UML diagrams.

The behavioural approach reviewed in this paper focuses on the dynamic aspects of program execution which can be converted easily into implementation codes but without much consideration on data processing. It involves simple notations for building diagrams that can easily be learned by any individual. The visual representation is clear and easy to be understood, referred to, and validated.
Table 1: Requirements Graphical Visualization Techniques Comparison

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Requirements Graphical Visualization Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3D</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>This approach involves a lot of processes and the requirements need to be specified in Z language.</td>
</tr>
<tr>
<td>Validation Support</td>
<td>This technique was aimed to help increase the efficiency of requirements validation process and has succeeded. The 3D animations are highly understandable, which makes the validation process easier as the results are simulations of the real world.</td>
</tr>
<tr>
<td>Comprehensiveness</td>
<td>This technique seems to be relevant for any functional requirements that might need simulation such as the elevator or automated teller machine system.</td>
</tr>
<tr>
<td>Ease of Conversion to Implementation Codes</td>
<td>Users of this technique need to specify their requirements in a formal language, Z. Therefore, converting this formal language into codes should be quite easy.</td>
</tr>
</tbody>
</table>
4 Conclusion
A comparative analysis approach based on four important attributes was done. Each technique has its strengths and weaknesses. More techniques are required to be developed through research to broaden the number of techniques that can be chosen from.

It seems that there are a lot of requirements visualization techniques around, but not many focus on representing the requirements visually themselves. Future work includes the development of a new visualization technique that will support visualization of a variety of requirement types and aspects.

References: