Visualizing Symbolic Transition Systems

Dennis van der Werf

Spring 2018, 44 pages

Supervisor: Robert Belleman and Machiel van der Bijl
Host organisation: Axini, http://www.axini.com
# Contents

Abstract 3

1 Introduction 4  
1.1 Motivation .................................................................................. 4  
1.2 Research Question ...................................................................... 5  
1.3 Contributions .............................................................................. 5  
1.4 Research Method .......................................................................... 5  
1.5 Outline .......................................................................................... 5  

2 Background and related work 6  
2.1 Model-based testing ...................................................................... 6  
2.1.1 Labelled transition systems ......................................................... 6  
2.1.2 Symbolic transition systems ......................................................... 7  
2.2 Visualization .................................................................................. 8  
2.2.1 Information visualization ............................................................ 9  
2.2.2 Graph visualization ................................................................... 10  

3 Original situation 13  
3.1 TestManager ................................................................................ 13  
3.2 Task Analysis .............................................................................. 14  
3.2.1 Use cases .................................................................................. 14  
3.2.2 Design focus ............................................................................. 15  

4 New visualization method 16  
4.1 Design .......................................................................................... 16  
4.1.1 Visual Mapping ........................................................................ 16  
4.1.2 Additional functionalities .......................................................... 19  
4.1.3 Technologies chosen ................................................................. 19  
4.1.4 Architecture .............................................................................. 20  
4.2 The Visualization .......................................................................... 21  
4.3 Functionalities .............................................................................. 22  
4.3.1 Semantic zooming .................................................................... 23  
4.3.2 Search panel .............................................................................. 25  
4.3.3 World-in-miniature ................................................................... 25  
4.3.4 Detail panel ................................................................................ 25  

5 Evaluation 26  
5.1 User Performance ........................................................................ 26  
5.1.1 Experiment Design .................................................................... 26  
5.1.2 Results ...................................................................................... 27  
5.2 User Experience ........................................................................... 29  
5.2.1 Results ...................................................................................... 29  

6 Discussion 31  
6.1 Experiment observations .............................................................. 31
Abstract

We introduce a new method for visualizing Symbolic Transition Systems (STS). Symbolic transition systems are state machines that are used for model-based testing. These, usually large, state machines contain a lot of information and are hard to comprehend with only a textual representation. Visualizing an STS improves its comprehensibility as it allows for faster and more efficient processing of information. However, visualizing an STS has proven to be challenging as the visualization would either be too big to comprehend or incomplete. Therefore a new visualization method is needed.

The method presented in this thesis is created by applying information visualization theory using a visual mapping that maps the most important information to the most expressive visual variable. Furthermore a technique called semantic zooming is used. Semantic zooming allows us to create a single visualization which provides both an overview of the entire structure and more detailed views when needed. To find out if our new visualization method is an improvement compared to the state-of-the-art, the following research question is answered: Is semantic zooming a more efficient technique compared to the state-of-the-art for exploring symbolic transition systems? A visualization is considered more efficient when users are able to identify and process information from an STS quicker.

Our new method is implemented in a proof-of-concept visualization tool. This tool is evaluated and compared to the old, currently in use, visualization method using a questionnaire and a series of experiments carried out by domain experts. The results of these experiments indicate that our new visualization method is able to visualize an STS more efficiently than the state-of-the-art.
Chapter 1

Introduction

A well-known rule of thumb is that around 50% of the time and budget of a software project is spent on software testing [20]. It is therefore a major part of the software development process and an important step that ensures that a system has a certain level of quality. Tests are often executed either manually or automatically. Manual testing often requires significant effort as it is necessary to repeatedly execute the same tests. This makes them error prone and is a good reason for automating them. Automated tests reduce the effort it takes to run the test but bring additional costs upfront as the test scripts need to be developed and maintained.

Another approach is model-based testing (MBT), which is a formal testing technique that automates the test case generation, test execution and test evaluation. MBT generates test cases based on a model of the system under test (SUT). This model formally describes the behaviour of the SUT and can therefore provably validate if it conforms to its specifications [33].

The behaviour of the SUT is often modelled using a variation of state transition systems namely symbolic transition systems (STS). These transition systems contain all the states of an SUT and all the transitions/actions it can perform. Large or complex SUTs often result in enormous transition systems. As these systems grow, understanding them using only a textual representation becomes increasingly harder. To counter this problem the transitions systems can be visualized. Visualizing information allows the user to process it more effectively and efficiently [34]. However, the challenge still remains. These transitions systems usually contain too much information to visualize everything at once.

1.1 Motivation

TestManager is a tool, created by Axini, to facilitate model-based testing. This tool allows users to create models, generate, execute and analyse test cases and visualize models. The visualization used in TestManager encounters the same challenge as mentioned before and is therefore only able to show small parts of the model at once. A big drawback of this method is that the context in which that small part of the model resides is not visualized. Our research aims to solve this issue by dynamically changing what information of the model is visualized, providing the user with both an overview of the entire model and detailed information when needed. What is visualized is based on the current zoom level. This means that users will be able to zoom in on specific parts of the model which will gradually reveal more details along the way. This is also known as semantic zooming.
1.2 Research Question

This thesis aims to find an answer to the following question:

Is semantic zooming a more efficient technique compared to the state-of-the-art for exploring symbolic transition systems?

Where “more efficient”, in this case, means that users are able to identify and process information from a model more quickly compared to the state-of-the-art. The visualization used by Axini is considered as the state-of-the-art. To answer this question, we first need to find an answer to the question “What information needs to be communicated to explore an STS?” Once we know what information needs to be communicated we need to find out what the most effective way of visualizing this information is. We do this by answering the question “What visual variables are most suitable to encode what aspects of an STS?” Then we research if this information can be communicated effectively using semantic zooming. This prompts the following questions: “Does semantic zooming improve navigating a visualized STS?” and “Does semantic zooming help provide the necessary context, while exploring a visualized STS?” Answering these questions will provide us with the necessary insights to formulate an answer to the main research question.

1.3 Contributions

This thesis presents the following contributions:

1. A task analysis that breaks down what information users expect when inspecting a model.
2. The rationale behind the visual encoding of the elements contained by a model.
3. A tool showcasing a proof-of-concept implementation of the new visualization method.
4. An experiment that compares the new visualization method with the traditional visualization.

1.4 Research Method

First a task analysis is performed. Input for the task analysis is collected by sitting down with three domain experts, asking them about their daily tasks, to demonstrate it while thinking aloud and sharing their thoughts about the current visualization. This is also known as apprenticing [2]. With this information we want to get a good understanding on how the current visualization is used, what information they are looking for and where the current visualization is inadequate for the task. Then a visual mapping will be made where the most important information is mapped to the most expressive visual variable.

Based on this information and the findings from the task analysis we develop a proof-of-concept that uses semantic zooming to visualize the required information. Then we collect, with the help of domain experts, both quantitative and qualitative measures to compare the proof-of-concept with the original visualization. We collect quantitative measurements using a series of experiments and a questionnaire provides the qualitative measures.

1.5 Outline

This thesis is structured in the following manner. Chapter 2 provides an overview of the background and related work on model-based testing and information visualization. The original visualization, that is currently in use, and a task analysis is presented in chapter 3. Chapter 4 presents the rationale behind our new visualization method, the new visualization method and its functionalities. The new method is evaluated in chapter 5, where we measure both the user performance and user experience. In chapter 6, the results of the evaluation are discussed. Finally, conclusions are drawn and future work is suggested in chapter 7.
Chapter 2

Background and related work

This chapter covers and describes the relevant and related work for this research to make it self-contained. It provides the context in which this research is conducted and discusses the relevant work in the two relevant research areas: Model-based testing and Information visualization.

2.1 Model-based testing

Model-based testing is a software testing approach where test-creation, test-execution and result analysis is automated. Automating the test creation step requires a formal model of the system under test (SUT). This is commonly done using labelled transitions systems. Axini uses symbolic transition systems that are based on labelled transitions systems. Tretmans describes model-based testing as “formal, specification based, active, black-box, functionality testing” [33]. Formally because the desired behaviour is defined in a formal language, there is a formal definition of what a conforming SUT is and a correctness proof of the generated tests. Specification based because the expected behaviour is specified in a model. Active because the test tool controls the SUT in an active way using stimuli and observing responses. Furthermore, the tests are black-box because the test tool only accesses external interfaces of the SUT without knowing its internal structure. [33]

2.1.1 Labelled transition systems

A labelled transition system (LTS) is a structure consisting of states and transitions. [33] These states and transitions represent the behaviour of a system. Depending on what state the system is in it can perform certain actions. An action can be either an input or an output action.

A labelled transition system is defined as a four-tuple \((Q, L, T, q_0)\) where

- \(Q\) is a countable, non-empty set of states.
- \(L\) is a countable set of input and output labels. Outputs are actions initiated by the system and inputs are initiated by the environment.
- \(T \subseteq Q \times (L \cup \tau) \times Q\), with \(\tau \not\in L\) is the transition relation where \(\tau\) is a special label used for actions that are unobservable to the outside world.
- \(q_0 \in Q\) is the initial state.

Figure 2.1 shows an LTS of a simple coffee dispenser. Inputs are denoted with a question mark and outputs with an explanation mark. This coffee dispenser accepts coins of 10, 20 and 50 cents and the cost of coffee is 50 cent. If the user inserts a total of 50 cents and presses the button, the machine will dispense coffee. As we can see there are already quite some states to model this simple behaviour. Now imagine that the behaviour is extended and the system now accepts all coins, allows the user to input more than 50 cents to get multiple cups of coffee and afterwards returns the change. The number of states and transitions will grow enormously. The need to model each data value separately can lead to what is known as a state space explosion. In a more complex system, with unbound data domains, the number of states can theoretically grow infinite.
2.1.2 Symbolic transition systems

As complex systems often use large or unbound data domains, a state space explosion becomes a common problem. To counter this problem, the definition of an LTS is extended by introducing state variables, label parameters and update functions resulting in the following definition:

A symbolic transition system is defined as $\langle L, l_0, V, \iota, I, \Lambda, \rightarrow \rangle$ [5] where

- $L$ is a countable set of states.
- $l_0$ is the initial state.
- $V$ is a countable set of state variables. These are global variables, accessible from the entire model and are used to store data.
- $\iota$ is an initialisation of the state variables.
- $I$ is a set of label parameters, disjoint from $V$. These parameters are local to their transition and not accessible from the outside of the transition in which they are used.
- $\Lambda$ is a set of observable transition labels.
- $\rightarrow$ is the transition relation. Each transition has either a label from $\Lambda$ or is unobservable and therefore labelled with $\tau$. Transitions can have an update mapping and a constraint. A constraint is a logical formula using variables from $V$ or $I$. If the constraint is resolved to true than the transition can be followed. If the transition is followed and it has an update mapping then the mapping is applied. This update mapping is a set of assignments that will update values in $V$.

By treating data symbolically it is no longer necessary to have separate transitions to represent different data values. Figure 2.2 illustrates a similar coffee dispenser as in the previous section. Since data is treated symbolically there is no longer a need to have a transition for each different data value.
This results in much fewer states and a compacter model. The first transition, from state 0 to state 1 is labelled with an input where the user can insert a coin. The coin can have any value greater than 0 and its value is added to a balance (money). If the balance is less than 50 then it returns to state 0 waiting for more coins. If after inserting a coin the balance is equal or greater than 50 than the system will wait for the button_pressed input. When this input is received the system will dispense coffee and lower the balance with 50. This also means that when the system will accept all coins the STS will stay the same, eliminating the state space explosion. Adding an additional transition between state 3 and state 0 would allow to get multiple cups of coffee.

2.2 Visualization

The importance of the visual notation of information in software engineering is often overlooked [18]. Visualization can be a great tool to obtain insights in data in an efficient and effective manner. This is all thanks to the unique capabilities of the human visual system [34]. Around a quarter of our brain is devoted to this system, which is more than all other senses combined [18]. Using visualization to transfer information also has several other benefits: Diagrams can convey information more concisely [4] and more precisely compared to ordinary language [15], it enables us to detect interesting features and patterns in a short period of time [35] and it is more likely that the information is remembered due to the picture superiority effect [8].

Despite all these benefits, why is the visual notation of information not always considered important? Moody mentions three possible reasons in his paper [18]. He states that researchers see visual notations as informal and that the methods for analysing visual representations are less mature compared to methods for analysing verbal or mathematical representations. However, Harel and Rumpe conclude that visual notations are not less formal than textual ones [10]. A third explanation is that software engineering researchers simply consider visual notation unimportant. Decisions about the visual syntax are often considered trivial or irrelevant, a matter of aesthetics rather than effectiveness [12]. However this is not true, there are at least three empirical studies that show that the form of representation significantly affects understanding [25] [36] [21]. Furthermore, creating an effective visualization is hard and not trivial. It requires knowledge on how humans process visual information and how this can be used to effectively encode data. This will be discussed in the next section.
2.2.1 Information visualization

The transfer of information consists of three processes: encoding, decoding and interpretation.[18] Encoding takes place in the design space while decoding takes place in the solution space. The intended message (information) is first encoded into a diagram after which the receiver (user) will decode the diagram in a message which is hopefully the same as the intended message. Just like verbal communication these steps are both subject to “noise”.

There are eight visual variables that can be used to graphically encode information [1]. The variables that can be used are: shape, size, colour, brightness, orientation, texture, horizontal position and vertical position (illustrated in figure 2.3). The encoding has a big impact on how efficient and how effective a message can be decoded. The advantage of visualization is not that it can contain more information but that indexing this information can be done in an extremely efficient way [15]. This is thanks to how the information is processed. It happens in two phases. Perceptual processing (seeing) which is automatic, very fast and executed in parallel and cognitive processing (understanding) which requires attention, is relatively slow, effortful and sequential [18].

When information is encoded, it is important to use a suitable visual variable that is effective for that type of information. There are three types of data [31].

1) **Nominal** data which contain items that are distinguishable but that cannot ranked (e.g. types of fruit or gender),
2) **Ordinal** data which have a explicit ordering (e.g. army ranks or education levels) and
3) **Quantitative** which is numeric and therefore has not only an order but the distance between items can be calculated as well.

Using “colour” to encode quantitative data is much less effective than using it to encode nominal data. The same applies to, for example, “size” which is a highly effective encoding for quantitative data but less effective for ordinal and nominal data [13] [19]. For example using the colour blue to represent a smaller value and the colour yellow for a bigger value is considered bad practice. Blue is not naturally interpreted as “more” than yellow. A better variable would be size as a “bigger” circle is interpreted more naturally as a bigger value. This indicates that not all visual variables are equally effective in encoding information. Mackinlay and Jock proposed a conjectural theory where they ordered the visual variables based on their effectiveness. They also added several other visual variables. Figure 2.4 shows the effectiveness of all variables depending on the type of data.

The combination of the chosen variables to encode information is considered the “primary notation”. Visualizations usually also have a “secondary notation” which refers to visual variables that are not formally specified [22]. For example, size and position carry no official meaning in UML diagrams but can (unintentionally) convey information. When the secondary notation unintentionally distorts the intended message than this is considered “noise”. To counter noise and improve the discriminability of visual elements, a redundant coding can be used. Redundant encoding is the use of multiple visual variables for the same information [18].

---

1 Ratio and interval can be grouped as quantitative in this context.
Additionally, there is a phenomena that should be taken into account during the analysis and development of visualization methods. Studies on visual memory and change detection revealed a surprising inability to detect changes to scenes from one view to the next. For example, observers would most of the time fail to notice when the central actor in a motion picture was replaced by another actor wearing a different outfit [28]. This is also known as “change blindness”. Change blindness usually occurs after a brief visual disruption such as eye movement or a flashed blank screen.

In general there are two visualization categories [34]. The first category is exploration. In this case the user usually does not know what is in the data and uses the visualization to analyse and get a better understanding of the visualized data. The other category is presentation. In this case the visualization is used to communicate known data or information to other users. This can, for example, be a UML-diagram or a pie chart.

There is, especially for explorative visualizations, a well known and highly cited mantra of what a visualization should adhere to. “Overview first, zoom and filter, then details-on-demand” [27]. This mantra summarizes how to achieve rapid, user-controlled, information exploration. First it is important to place it into context by providing an overview. Then the user can filter out uninteresting items and zoom in on items of interest. Once the user has selected an item or group of items, details should be presented [27].

### 2.2.2 Graph visualization

The symbolic transition systems that Axini uses to model software behaviour have a very similar structure compared to directed graphs. Therefore the methods used to visualize a directed graph may also be of interest for visualizing the symbolic transition systems. This also applies for the challenges that arise when visualizing a graph.

Large graphs usually contain large numbers of nodes and edges. This makes it challenging to have a layout that displays it in a comprehensible way. The visualization can become dense, can have lots of crossing edges and overlapping elements [11]. There are several ways to handle these challenges ranging from specialized layout algorithms to abstracting the graph data to hiding less relevant data.
Clustering

A popular technique to counter dense graphs is by clustering groups of nodes. This helps reducing the number of elements by abstracting them based on certain attributes or properties. A cluster can also be created using other clusters. Figure 2.5 illustrates how a graph can be clustered.

Layout algorithms

There are several other popular graph layout algorithms. Many are based on the work of Sugiyama, Tagawa, and Toda [32]. They proposed a four step method designed for hierarchical structures. This method also aims to reduce the number of edge crossings. Built upon this work is an algorithm designed by Gansner et al. [7] which optimized this method. This algorithm is created to come up with effective layouts for directed graphs and is therefore of high interest for this project.

Visualization approach

A very related, topology focused, approach is that proposed by Meesters [17]. This research was also conducted using the Axini toolset. A visualization consisting of two side-by-side panels is presented. The left panel is used as a presentative panel which shows an overview of the STS. Once a selection is made in the left panel the right panel will show a more detailed view of that selection. This makes the right panel a explorative panel. The right panel can also be used for two other purposes. It can show test case traces and search results. Domain experts had mixed feelings about this approach because the visualization became to small due to the screen being split in half [17].

Figure 2.6 shows the visualization created by Meesters. On the left panel a number of states are selected. These selected states are then visualized on the right panel with additional information such as transition labels.
Figure 2.6: The visualization proposed by Meesters. The left panel shows an overview of the model. The user is able to decide and select which states are of interest. The selected states are then visualized on the right panel with additional information [17].

**Semantic zooming**

Semantic zooming is a form of details-on-demand that lets the user see different amounts of detail in a view by zooming in and out. “With a conventional geometric zoom, objects change only their size. Semantic zooming allows objects to change shape, details (not merely size of existing details) or their presence in the visualization, with objects appearing/disappearing according to the context.” [29]. This can for example mean that when the user is zooming out, a group of nodes and edges can be replaced by a single node or when zooming in object labels become visible.

A well known implementation of semantic zooming is its use in visualizing world maps. In these implementations users can both get an overview of the entire planet while also being able to zoom in all the way to street level.
Chapter 3

Original situation

This chapter describes the original visualization in TestManager. Furthermore it will discuss the limitations and other findings that were discovered during the task analysis. First the original visualization, that is currently used in TestManager, will be discussed. Second the task analysis will be described.

3.1 TestManager

TestManager is a tool developed by Axini that provides all the necessary functionalities to model and test software systems. These software systems are modelled using the Axini Modelling Language. This language makes use of the STS theory and added abstractions to create a more hierarchically structured model. These abstractions are processes and behaviours. Processes are the highest form of elements and usually contain one or more behaviours. Behaviours are sub-routines that allows similar parts of the system to only be modelled once.

The current visualization in TestManager is a directed graph based on the work of Sugiyama, Tagawa, and Toda. It is, just like the rest of TestManager, rendered in a web browser. The visualization is topology focused where the first state of a model is visualized at the top. TestManager is only able to visualize a single behaviour at a time. An example of a visualized behaviour can be seen in figure 3.1(A). The visualization contains three types of elements: ‘states’, ‘transitions’ and ‘behaviours’. States are represented with an oval containing the name of the state in the middle of the element. Transitions are represented with curved lines with a arrow head pointing at the target. The transition labels are drawn next to these lines. The rectangles in figure 3.1(C) are behaviours. They contain the name of the behaviours and are clickable. When the user clicks on a behaviour a new tab opens in the browser and the content of that behaviour is shown. These different tabs can be seen in figure 3.1(A). Users are also able to switch to an overview, which also opens a new tab and visualizes all non-terminating behaviours and how they are connected. This can be seen in figure 3.1(B).

Furthermore the visualization can be used to visualize test cases. In this case the model is visualized in the same way as explained above but it will, in addition, use colours to mark the sequence of states and transitions that are executed in this test case.
3.2 Task Analysis

To get a good understanding on what domain problem the visualization should solve, what tasks are performed using the visualization and what the limitations are of the current way of visualizing the model, a task analysis was performed. This task analysis consisted of apprenticing with three domain experts [2]. We asked them about their daily tasks, to demonstrate it while thinking aloud and sharing their thoughts and experiences with the visualization.

3.2.1 Use cases

The task analysis revealed that the visualization is used in three use cases. Our proof of concept will address the most common use case.

Modelling - The first and most common use case is during modelling. When users are modelling the behaviour of an SUT, they will use the visualization to explore the model that they are creating. The main purpose of this is to verify that the visual representation of what they are modelling matches their expectations. The most notable quote was from one of the experts who said: "The visualization is like a compiled version of the model". The main action they do to perform this task is navigating through the model towards the part that they are working on. This is also the main issue they express...
regarding the current visualization. To counter the visualization challenges mentioned earlier, TestManager is only able to visualize small parts of the model. This makes navigating through the model cumbersome and time consuming. Every time the user navigates to the next part of the model, a new tab opens in the browser. Every time a new tab opens, change blindness occurs. The tab takes a few seconds to load, the user has to remember and recognize the part of the model they are now looking at and then decide on the next step they need to take to get to the part they want to be. Once they are at the point of interest, the context is missing. As only a small part is visualized, it is not always clear how this part relates to the rest of the model or how the states in this part of the model can be reached.

Other use cases that were identified, but disregarded for the rest of the research, are test case analysis and coverage analysis. Test case analysis is usually performed when a test case fails. In this case the visualization highlights the trace of the test case and indicates where it failed. During coverage analysis the user wants to find out which states and transitions are covered by a test set. The visualization visualizes the model and indicates what parts of the model are covered, partially covered or not covered.

3.2.2 Design focus

During the task analysis of the most common use case, two main issues were identified. Navigating the model is regarded as time consuming. Users often knew what part of a model they wanted to examine, but to get to the relevant part they often had to go through and process multiple irrelevant parts of the model.

The other main issue is the lack of context. Since only a part of the model is visualized at once, the user is required to memorize and mentally relate the other relevant parts of the model.

Both of these issues are directly related to the fact that the model is only partially visualized. To counter this, a single visualization containing the entire model is desirable. However when the entire model is visualized at once, the original challenges of visualizing large graphs will reappear. Therefore additional measures need to be taken. The visualization method presented in this thesis adheres to the following principles:

1. **Enable the user to explore the entire model in a single visualization.**
   Visualizing the entire model in a single visualization eliminates the loading time when a new part of the model is visualized. This in turn prevents the user from losing context and orientation that currently occurs when switching between visualizations. This also provides the user with an overview of the general structure of the graph.

2. **Reduce complexity by hiding irrelevant information**
   Because the model is now visualized in its entirety, measures are taken to reduce its cognitive complexity. This is done by hiding information depending on the context. And in our case the context is determined based on how far a user zoomed in on the graph structure. This allows for a comprehensible overview of the model and a more detailed view when needed.

3. **Use a visual encoding that allows for efficient processing**
   Hiding irrelevant information on its own is not enough to create a visualization that can be processed efficiently. Therefore a visual encoding is used that allows the user to process the visualized information in an efficient way. This can be done by mapping the most effective visual variables to the most important information in the model.

4. **Provide interaction that allows the user to get details-on-demand**
   Since the visualization automatically hides information it is important that the user can get the details-on-demand. To achieve this the user must be provided with mechanisms that allows them to obtain the required information. They have to be enabled to freely examine the model, focus on points of interest and get detailed information on demand. This will ensure that the visualization will conform to the “Overview first, zoom and filter, then details-on-demand” mantra.
Chapter 4

New visualization method

This chapter describes the design process and an implementation of the new visualization method. A tool is created as a proof of concept which will later be used to evaluate our new method. First the design will be discussed. A visual mapping is made and the appropriate technologies are selected. Then the tool and its functionalities will be presented and discussed.

4.1 Design

This section describes the design process of the proof-of-concept. A visual mapping is created and the choices that were made are explained. Furthermore the functionalities of the proof-of-concept and their rationale are described.

4.1.1 Visual Mapping

As explained in section 2.2.1, not all visual variables are equally effective in encoding information. Therefore a visual mapping has to be made. This mapping maps the visual variables to the information that needs to be visualized. First the information that needs to be visualized should be categorized by their data types (either quantitative, ordinal or nominal). Then, by prioritizing the information, a mapping can be made with the help of figure 2.4 in section 2.2.1. Creating this mapping ensures that the most effective visual variable is used for the most important information. To improve the discriminability of elements, a redundant coding is applied when possible. And since the new visualization method will implement semantic zooming, it is possible that visual variables will become available after a change in zoom level. Therefore, a visual mapping for each semantic zoom level could be made. However, in this case the mappings would be identical and thus only a single mapping is made.

The visual mapping of an STS is made in table 4.1. This table shows the different variables that are contained in an STS, a description of what information they carry, the data type of the variable and the assigned visual variable. These variables are ordered by importance where state order is the most important and behaviour size the least important information.

**State order** - As STSs are essentially directed graphs, the order of the states is the most important property when visualizing an STS. This is also important when the user wants to understand the structure of the STS. E.g. an STS has a start state followed by one or more states that can be reached from the start state. Each of these states have their own reachable states, etc. Position is the most effective visual variable for ordinal information and is therefore assigned to this variable.

**Transition order** - Position is used to visualize the state order resulting in a node-link diagram. This makes that the position of transitions is constrained by the position of its related states. Therefore position is used here as well.
Table 4.1: The complete visual mapping of an STS. The variables are ordered by their importance in descending order.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Type</th>
<th>Visual variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>State order</td>
<td>This variable reflects the relation between states. E.g. the successors and the predecessors of the current state.</td>
<td>Ordinal</td>
<td>Position</td>
</tr>
<tr>
<td>Transition order</td>
<td>This variable reflects what possible actions can be taken in a certain state.</td>
<td>Ordinal</td>
<td>Position</td>
</tr>
<tr>
<td>Behaviour states</td>
<td>This variable reflects in which behaviour a state is included.</td>
<td>Nominal</td>
<td>Colour hue &amp; Containment</td>
</tr>
<tr>
<td>Transition type</td>
<td>This variable describes the transition type. E.g. if it is observable or not</td>
<td>Nominal</td>
<td>Texture</td>
</tr>
<tr>
<td>Behaviour type</td>
<td>This variable describes the behaviour type. E.g. if it is a terminating or non-terminating behaviour.</td>
<td>Nominal</td>
<td>Containment &amp; Saturation</td>
</tr>
<tr>
<td>State type</td>
<td>This variable reflects if the state has a transition to another behaviour.</td>
<td>Nominal</td>
<td>Shape</td>
</tr>
<tr>
<td>Behaviour size</td>
<td>This variable reflects the number of states in this behaviour.</td>
<td>Quantitative</td>
<td>Area</td>
</tr>
</tbody>
</table>

**Behaviour states** - To provide additional context it should be possible to recognize which states belong to which behaviour. This is also nominal information. Since Position is already taken colour hue is selected. Colour hue is a suitable variable as it allows for a great range of unique values without cluttering the visualization. Furthermore, containment can be used as a redundant coding without conflicting with other visual variables.

**Transition type** - The type of a transition is nominal information and can be encoded using texture without any conflicts. Since there are only two transition types the set of distinct textures will also be limited to two.

**Behaviour type** - A behaviour can either be terminating or a non-terminating. A terminating behaviour can be compared to a subroutine that after completion will return to the point from where it was initiated from. Whereas a non-terminating behaviour will just continue with the next behaviour. Therefore containment is used. Terminating behaviours will be contained within non-terminating behaviours. This also helps maintain context as the user is able to directly relate the terminating behaviour to the non-terminating behaviour. Colour saturation is used as a redundant encoding to further distinguish terminating behaviours from non-terminating behaviours.

**State type** - Shape is used to indicate that a state has a transition to another behaviour. This is nominal information and therefore shape is selected as that is the next most expressive variable.

**Behaviour size** - The behaviour size reflects the number of states that are inside a behaviour. This is quantitative information and as length, angle and slope conflict with position and therefore the layout of the graph, area is used.

A step-by-step application of this encoding can be found in figure 4.1. In figure 4.1(1) an STS without any encoded information is illustrated. The states are positioned randomly which makes it hard to see the order of the states. In figure 4.1(2) the order of the states and transitions is encoded using “position”. This already makes the STS much more comprehensible. Figure 4.1(3) shows which states belong to which behaviours by containing those states in a rectangle. This is further emphasized in 4.1(4) by giving the states in each behaviour different colours. Next the transition types are encoded in figure 4.1(5) using texture. Based on the texture of transitions it is now possible to see that the transitions towards “state 0” and from “state 1” are unobservable. Figure 4.1(6) illustrates how a terminating behaviour is contained in a non-terminating behaviour. The terminating behaviour is made more distinguishable by using colour saturation, which can be seen in figure 4.1(7). Figure 4.1(8) shows how “shapes” are used to indicate states with a transition to another behaviour. Furthermore it illustrates how the size of the behaviour indicates how many states it contains.
Figure 4.1: A step-by-step application of the visual mapping.
4.1.2 Additional functionalities

The new visualization focusses on mitigating the two main issues that were identified during the task analysis, the lack of context and time consuming navigation. The main strategy to counter these issues is to implement semantic zooming. Besides semantic zooming, three additional functionalities are developed to further mitigate these issues.

A search function will be added to enhance the navigation capabilities. With this search function users will be able to quickly locate states, transitions and behaviours. Furthermore a detail panel will be introduced, which will provide detailed transition information. This information provides additional context when the source or target of a transition is located beyond the bounds of the current view. In addition a world-in-miniature is added, which will provide the user with the context needed to relate the current view to the rest of the model.

4.1.3 Technologies chosen

During the development of the proof-of-concept, several decisions were made about what technologies will be used. This section discusses the integration with TestManager, what technologies were chosen and the rationale of these decisions.

TestManager Integration

TestManager is a web application and it uses the browser to render and visualize its models. This is done by exporting the model to a JSON object. Based on this JSON object, the visualization is created using Dagre-D3.1 Dagre-D3 is a javascript library that combines the Dagre2 layout algorithm and the D33 visualization library. As this JSON object contained all necessary information the decision was made to make use of the same JSON objects to gather the necessary information for visualizing the model.

This JSON object contains all the required information and there are no further dependencies on TestManager. Without these dependencies it is no longer necessary to integrate the proof-of-concept in TestManager. The proof-of-concept is developed using the same techniques that are used for TestManager but to speed up development, it is built as a stand-alone web application. The proof-of-concept is a client-server application built using Ruby on Rails. It serves a single page that renders the visualization.

Layout Algorithm

The visual mapping shows that the layout is the most important part of the visualization. The position of nodes is used to communicate the structure of the model and therefore force-directed layouts such as [6] and [30] and the attribute-based approach by Pretorius and Wijk [24] are not suitable. A popular directed-graph layout algorithm is Dagre. It is based on the work of Gansner et al. [7] and is currently used in TestManager as well. During the task analysis, users indicated that this layout algorithm produces a layout that is a good representation of the model structure. Therefore the decision was made to use this algorithm to encode the state and transition order.

Javascript Library

There is already a wide arrange of tools available for visualizing graphs. To reduce the development effort required for the proof of concept, an existing library was used. Four libraries were compared based on four criteria. The criteria were: Dagre, dynamic styling, world-in-miniature, extensibility. First the Dagre layout-algorithm must be supported, then dynamic styling is required to implement semantic zooming, the existence of a world-in-miniature is evaluated and an arbitrary rating between

1 https://github.com/dagrejs/dagre-d3
2 https://github.com/dagrejs/dagre
3 https://d3js.org/
1 and 5 is given for its extensibility. This rating is based on the available API, the documentation and the size and activity of the community. The library is rated with a 5 if these aspects are favourable. A partial supported criteria means that the functionality is not supported out-of-the-box but it can be developed using the API. Figure 4.1.3 shows the summary of this comparison.

<table>
<thead>
<tr>
<th>Library</th>
<th>Dagre</th>
<th>Dynamic styling</th>
<th>World-in-Miniature</th>
<th>Extensibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cytoscape.js</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>5</td>
</tr>
<tr>
<td>Dagre-D3.js</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>4</td>
</tr>
<tr>
<td>Sigma.js</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>3</td>
</tr>
<tr>
<td>Vis.js</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>2</td>
</tr>
</tbody>
</table>

✓ = Supported
✓ = Partial support
✗ = Not supported

Figure 4.2: Criteria overview per visualization library.

*Cytoscape.js* is selected as it is the most suitable solution for this use case. It provides the functionalities that are needed, is very well documented and has an active community.

Search panel

For the search panel a typeahead javascript library⁴ is used. This library is highly customizable and is able to search and suggest multiple types of objects at once. This is necessary as in this case it needs to search and suggest transitions, states and behaviours.

Other Consideration

Visualizing transition systems is not new and quite some research is done in how to visualize them effectively. Ham, Wetering, and Wijk [9] and Ploeger and Tankink [23] made a big contribution with their papers on interactive visualization of state transition systems. They dealt with the lack of visualization space by clustering states and using 3D visualizations. However, although this visualization gives an insight into large state spaces, it also comes with some limitations. Interpretation of these visualizations requires highly skilled experts and it sometimes raises more questions than it answers [9]. Furthermore, effectively incorporating important information such as behaviours and transition labels would be challenging. Therefore the decision was made to not proceed with a 3D visualization.

An example of a state transition system visualized using this method can be found in appendix A

4.1.4 Architecture

The proof-of-concept is developed as a Ruby on Rail web-application. The server-side of the application serves a single page that contains the full visualization and the modules for the additional functionalities. Figure 4.3 illustrates how the different modules are structured and what their function is. The JSON-converter retrieves the JSON-file containing the model from the server and converts it to a Javascript object. This object is then used, together with a stylesheet, by the “main” module to initialize the graph using Cytoscape. Cytoscape is extended with two extensions for the Dagre layout and the world-in-miniature. The “main” module also tells Cytoscape to center a certain element when

---

⁴https://github.com/running-coder/jquery-typeahead/
the user selects a suggestion from the search panel or clicks a source/target state in the detail panel. The reverse is also true, if Cytoscape detects that a user is hovering a transition, it tells “main” to show its information in the detail panel.

4.2 The Visualization

The visualization itself is a single page that contains the entire model and other functionalities. Figure 4.4 illustrates the starting point of the visualization. A model is visualized and we see how the behaviours relate to each other. Furthermore we can see additional functionalities, which are explained in section 4.3.

Figure 4.4 shows the starting point of the visualization. At this point the user is able to start navigating the model. They can do so by zooming in or clicking on a behaviour or they can start querying the model using the search functionality.
4.3 Functionalities

Interaction is an important part of a visualization. The main interaction that this research focusses on is semantic zooming. However, the task analysis showed that the visualization could benefit from additional functionalities. These functionalities will help the user orientate and provide context when navigating through the visualization.

An overview of the additional functionalities is illustrated in figure 4.5. (A) is the search functionality that is discussed in section 4.3.2, (C) is the detail panel that is discussed in section 4.3.4 and (D) is the world-in-miniature as explained in section 4.3.3.

Furthermore figure 4.5 shows a model on the “behaviour overview” zoom level (see section 4.3.1). We see two non-terminating behaviours (and a third one under the world-in-miniature) which we identify by the blue bounding boxes. In each behaviour we see a number of states and transitions where the states in each behaviour have a different colour. Figure 4.5 (B) shows a terminating behaviour that is contained within a non-terminating behaviour. This can easily be identified using four visual cues: the states are coloured differently, the behaviour bounding box has a darker colour, it is contained within another behaviour and the state that has a transition that goes to the start state of the terminating behaviour is rectangular.
4.3.1 Semantic zooming

Semantic zooming automatically hides, reveals or changes the appearance of information based on the zoom level of the user. In this case the visualization distinguishes three levels of detail. First the user is presented with an overview of the entire STS. When the user zooms in on a point of interest the visualization automatically reveals more information. The three levels of detail are also illustrated in figure 4.6.
Figure 4.6: The three levels of details distinguished by semantic zooming. (A) shows the *model overview*. (B) shows the *behaviour overview* of the “main” behaviour and (C) illustrates a *detailed view* of some states and transitions from the “main” behaviour.

**Model overview**

The model overview level show an overview of the entire model. See figure 4.6(A). It shows how behaviours are structured and connected. The behaviours are shown as rectangles with the behaviours name in it. States and transitions are less relevant on this level and are therefore made less visible to reduce the cognitive complexity of the graph. Transitions that have the source state in one behaviour and the target state in another are replaced by a larger single transition between those behaviours to indicate that they are connected. If there are multiple transitions between two behaviours they are bundled and represented as a single transition.
**Behaviour overview**

The behaviour overview shows how states and transitions are structured inside a behaviour. At this level, states and transitions are visible but transition-labels remain hidden to minimize the cognitive complexity and reduce clutter. This can be seen in figure 4.6(B).

When the user zooms in on a behaviour and therefore moves from the “model overview” to the “behaviour overview”, the name of the behaviour and the large transitions between behaviours are hidden. The user is now able to see how states are connected with each other. Furthermore, the user is able to get the label of a transition by either zooming in further or by hoovering the transition, which will show all the transition information in the detail panel (See section 4.3.4).

**Detailed view**

The detailed view shows everything that is visible in the behaviour overview plus the transition labels. This can be seen in 4.6(C). Using the transition labels, the user is able to examine the behaviour that is described by the model.

**4.3.2 Search panel**

In the case that the user already knows what information they are interested in, for example a certain state or behaviour, the search functionality helps them to quickly locate and navigate towards the relevant element in the visualization.

It contains a suggestion engine that suggests possible elements while the user is typing. The search panel suggests transitions, states and behaviours and matches the user input not only with the name of the elements but also with other information such as transition labels. This allows the user to search for transitions that have a constraint on a certain variable. There is also a indication for the user to see where the input matches the element. Figure 4.5(A) illustrates an example of the suggestions when the user searches for “idle”. If the user selects a suggestion, the visualization centres that element using an animation. This is helpful in cases where the user already knows what element they are looking for but not where it is located. An animation is used which enables the user to maintain orientation. Without the animation the user would have to reorientate after the element is centered.

**4.3.3 World-in-miniature**

A world-in-miniature is implemented to provide additional context when viewing a part of the model. A world-in-miniature is a mini-map usually placed in the corner of the screen. This map shows the model overview with an indication of what part the user is currently looking at. This provides additional context and helps the user orientate when navigating on a lower level of detail.

Figure 4.5(D) illustrates the world-in-miniature. It shows the overview of the model as seen in figure 4.4. The orange rectangle indicates what part of the model is currently visualized. This also means that this indicator gets bigger or smaller when zooming in or out as more or less of the model is present on the screen. This can be seen in figure 4.6(A) where the entire model is present on the screen and figure 4.5(D) where only a small part is visualized. The world-in-miniature can also be used to navigate the visualization. Dragging the indicator or clicking somewhere in the world-in-miniature moves the view to that location.

**4.3.4 Detail panel**

The detail panel is used to get additional transition details on demand. This can be useful when, for example, a transition does not fit on the screen or the label is not visible because the user is on the behaviour overview zoom level. The user can access this information by hoovering a transition. This fills the detail panel with information about that transition. The detail panel shows the source state, target state and the transition-label. An example is shown in figure 4.5(C).
Chapter 5

Evaluation

Evaluating visualization methods is a complex and challenging process [3]. It is therefore not uncommon for visualization papers to not include an evaluation section. Although there is a trend that more and more papers start including some type of evaluation, Lam et al. found that only 361 papers of the 850 papers submitted to the EuroVis, InfoVis, and VAST conferences included evaluations [14].

We evaluate two scenarios. **User performance** and **user experience**. These two scenarios were chosen as they help us the most in finding an answer to our research question. Another scenario that is often used for evaluation is **algorithm performance** where for example **frames per second** or **memory usage** is measured and compared to other methods. This is not the focus of our research and is therefore not evaluated.

5.1 User Performance

To find out how the new method affects user performance, a series of experiments is conducted. These experiments should be as close to the day-to-day activities of the user as possible. And as visualization is a tool to get a better understanding of a model, the experiments should reflect this. The experiments are therefore questions about the behaviour of a model and not questions about the visualization itself. Instead of asking users to locate and navigate to certain points in a model, more work related questions are asked, e.g. “The system performed a certain transition and is now in this state, what could be the next transition?” Answering this question requires the user to not only navigate the visualization but also to understand, interpret and apply their model-based testing (MBT) knowledge to the data that is visualized.

User performance is measured in terms of time. Another measure that was considered was number of interactions. However, not all users interact with the visualization in the same way. For example, some users perform three small steps when navigating the model where another user may use a single large step to get to the same point. We therefore conclude that time is a more relevant measure than number of interactions.

5.1.1 Experiment Design

To test the impact on the user performance of the new visualization method, we asked eight domain experts ten multiple choice questions (See Appendix B.1). The domain experts have experience with MBT ranging from 1.5 to 15 years. The visualization is presented on a desktop pc. The multiple choice questions are presented and answered using a tablet. This tablet also does the time measuring. The user is able to read the question and if needed ask for clarification if the user does not understand the question. Once the user is ready, they press a start button on the tablet after which the possible answers appear. At this point, the user starts using the visualization to find the answer. When the user thinks they know the answer, they select it on the tablet after which the time is stopped. After each question the visualization is reset to make sure that all questions are answered using the
same starting point. The tablet ensures a uniform way of measuring with both the original and new visualization without having to switch windows when users want to reread the question or know the answer. Any overhead caused by the tablet such as the time between pressing start and actually starting and the time between finding the answer and selecting the answer is considered to be equal with both visualization methods.

The eight experts are split into two test groups of four, group A and group B. Both groups answer two questions using the old visualization to get a base time. Then group A answers half of the remaining questions using the old visualization and the other half using the new visualization. Group B answers them in the opposite way. Table 5.1 shows the question distribution and what visualization is used to answer it. This distribution is used as it is not possible to ask the same questions twice. Letting group A answer all the questions using the original visualization and group B using the new visualization would result in at least three factors influencing the results.

This distribution is used to counter the following three factors: The first one is MBT experience, where users with more MBT experience may be able to answer the questions faster. The second factor is model knowledge. Since an existing model is used, the user may be more or less familiar with the model. This could benefit them by using model specific knowledge to orientate. The third factor is the user’s experience with the original visualization. Some users may be able to use the visualization more effectively than others. This distribution causes these factors to influence the results for both the original and new visualization in an equal manner making them no longer a variable between the experiments.

Before starting the experiments, the user is asked to answer an example question. This allows the users to get familiar with the tablet and the type of questions. Another factor that could influence the results is that users have no prior experience with the new visualization method. Before answering the questions with the new visualization, an explanation is given. All features and functionalities are explained and the users are asked to answer the example question again so they get a feel with the new visualization. A more desirable approach would be that user have equal experience with both visualizations. Unfortunately, this was not feasible during this research.

### 5.1.2 Results

The eight domain experts produced four results per question per visualization. Given the small sample size we use the median of these results and not the mean. This decision was made because the median is more robust against outliers. Furthermore a delta is calculated to see if the question was answered faster or slower than the new visualization. Table 5.2 show all the medians and delta per question per visualization. Figure 5.1.2 show all measurements per question. The full results of each expert can be found in appendix B.2.
Table 5.2: The medians for each question in seconds. A negative delta indicates that the question was answered faster with the new visualization.

<table>
<thead>
<tr>
<th>Question</th>
<th>Median time to complete in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Original visualization</td>
</tr>
<tr>
<td>1</td>
<td>57</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>59</td>
</tr>
<tr>
<td>4</td>
<td>56</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td>117</td>
</tr>
<tr>
<td>7</td>
<td>102</td>
</tr>
<tr>
<td>8</td>
<td>52</td>
</tr>
<tr>
<td>9</td>
<td>40</td>
</tr>
<tr>
<td>10</td>
<td>102</td>
</tr>
<tr>
<td>Avg</td>
<td>70</td>
</tr>
</tbody>
</table>

Figure 5.1: All user performance measurements and its medians.
5.2 User Experience

Next to the quantitative experiments to measure the user performance, a questionnaire was used to evaluate the user experience. This questionnaire is used to evaluate how the domain experts react to the new visualization method. It measures, among others, perceived efficiency, satisfaction and features liked/disliked. The questionnaire consist of sixteen questions that are expressed with a five-point Likert Scale and three open-ended questions. These are the aspects covered by the questionnaire accompanied with their statements:

1. **Overall impression**: What is your overall impression of the new visualization method?
2. **Ease-of-use**: The visualization method is easy to use.
3. **Responsiveness**: The visualization provides a good user experience in terms of responsiveness.
4. **Perceived efficiency**: The visualization method will help me work more efficiently.
5. **Cognitive complexity**: The amount of data that is visualized at once can be overwhelming.
6. **Perceived correctness**: The layout of the states and behaviours are a good representation of the overall model structure.
7. **Information relevance**: The information presented on each level of detail is relevant to that level.
8. **Intuitiveness**: Changing between levels of details (zooming) is intuitive.
9. **Search function**: The search function provides useful suggestions (search results)
10. **Semantic zooming**: The semantic zooming does a good job in hiding irrelevant data.
11. **Detail panel content**: The detail panel helps me to get additional transition information when needed.
12. **Detail panel usefulness**: The detail panel is a useful addition
13. **Orientation**: I’m able to navigate the model without losing orientation.
14. **Interaction**: The visualization provides enough interaction to locate and navigate to points of interests
15. **World-in-miniature**: The world-in-miniature helps me to orientate.
16. **Encoding**: The visual encoding(colours, shapes, etc) helps me to get a better understanding of the model

The three open-ended questions were: “What are your thoughts on semantic zooming, is it a useful technique for this use case?”, “What are your thoughts about the search function?” and “Do you have any comments on the visualization in general?”.

The questionnaire is filled out by all eight domain experts after they concluded the user performance experiments. During these experiments they were able to get a impression of the functionalities and how it helped them in finding an answer to the experiment questions.

5.2.1 Results

The questionnaire proved to be a useful addition to the user performance experiments. It provided new insights in how the visualization is perceived and what functionalities are deemed effective. The results can be found in figure 5.2. The horizontal (blue) lines represent the full span of the given ratings and the vertical (red) lines represent the mean\(^1\).

\(^1\)It can be discussed if averaging an ordinal scale is appropriate, but we believe that, in this case, this is the best way to convey the results.
For the open-ended questions, all eight experts agreed that semantic zooming is a useful technique for exploring models. They also unanimously said that the search function was a very helpful and useful addition. The general sentiment towards the new visualization was very positive. Four experts explicitly said that it is an improvement compared to the current visualization and two experts said they would like to see it integrated in TestManager.

The complete results of the questionnaire can be found in appendix B.1.
Chapter 6

Discussion

6.1 Experiment observations

During the user performance experiment a number of observations were made where the questions influenced the performance. For example, question 4 asked what behaviours make use of a certain state variable. Three out of four experts who answered the question with the old visualization deducted from the multiple choice answers that they would only have to check a single behaviour to find the answer. If this was an open-ended question the user would have to manually check all behaviours. This is not the case with the new visualization method as it is possible to query the state variable with the search functionality.

Question 7 was another question where the question influenced the measured performance. The question describes a behaviour and a stimulus to give the user an indication on were to look in the model. Then the question states the response that is given and the question is what the next stimulus will be. Three experts ignored the first part of the question and started by querying the response. However in this case the given response occurs fifteen times throughout the model. As the necessary information to identify the correct one is not given by the search panel, they failed to locate the correct one. After some trial and error they decided to start over. When the users would first locate the start stimulus they no longer had trouble finding the correct answer.

6.2 Experiments conclusions

The results of the user performance experiment indicate that domain experts are able to answer questions about a model, on average, 37% faster with our new visualization method compared to the original visualization method.

The results show that five out of eight questions were answered faster, one was answered equally as fast and two were answered slower with the new visualization. When we take a closer look at the two questions that were answered slower we made some interesting observations. For one of the questions the search panel would suggest a transition with a very similar label compared to the one given in the question. This would put the users on the wrong track.

For the other question users were not able to interpret the search suggestions correctly. They would search for a state using the state name but would then select a transition from the suggestions. This would once again put them on the wrong track.

Considering our observation and the very limited time the users have spend with the new visualization, we believe that once the users become more familiar with the new visualization and have a better understanding on how to use it more effectively the results will shift in favour of the new visualization method. However, at this point this is nothing more than a hypothesis that requires additional validation.
The user experience questionnaire produced some interesting results as well. In general users were very positive. The majority of the users had a “very good” overall impression of the visualization and agreed that it would help them work more efficiently. They also agreed that the new visualization is intuitive and easy to use, responsive and provides enough interaction. For the visualized data they agreed that each level of detail provides relevant information, the layout provides a good representation of the model structure and the visual encoding helps them to get a better understanding of the model.

Functionality wise the users showed a little more disparity. They all agreed that the search functionality is a very useful and helpful addition but they had mixed feelings about the world-in-miniature. Half of the users agreed that it helped them orientate while the other half neither agreed or disagreed. On the detail panel the majority of the users were unsure if it is a useful addition. They also would neither agree or disagree that it helped them get additional information. This could be a interesting starting point for future work. Perhaps the detail panel takes up valuable space on the screen without showing enough relevant information.

Overall the new visualization method is considered a big improvement over the original visualization. The user performance experiments confirm that users are able to answer questions about models faster.

6.3 Threats to validity

The main threat of validity is the small number of domain experts used to evaluate the visualization method. This means that the conclusions cannot be generalized without further evaluation. Another threat to validity is the scalability and the structure of the models. Both of these aspects have an impact on the layout of the graph. During this research models where used that are, by Axini standards, average sized with a commonly used structure. Therefore further evaluation is necessary to verify that this is still a more efficient visualization method for uncommon structures. For example, models with a single large behaviour or a model with a substantial amount of small behaviours.
Chapter 7

Conclusion

7.1 Summary

In this thesis, we have presented a new visualization method for symbolic transition systems. We have shown that domain experts welcome this new method and that they are able to answer model-related questions more efficiently.

To develop this method we first conducted a task analysis, which showed that the most common use case for the visualization was during modelling. A visual mapping is created where we mapped the most expressive visual variable to the most important information. Additionally a technique called semantic zooming is used, which allows to visualize an entire model in a single visualization. As a result of this, users are now able to navigate and explore a model faster and without losing context.

The visualization offers three additional functionalities that enhances exploration and navigation. A search panel, detail panel and world-in-miniature are added. The detail panel and the world-in-miniature provide additional context and the search panel helps locating points-of-interests.

We have evaluated the new visualization method by measuring the user performance and user experience. The performance is evaluated by measuring task execution time and the user experience is measured with a questionnaire. Based on the results of the evaluation, we conclude that our new visualization method is able to visualize symbolic transition systems more efficiently than the state-of-the-art.

7.2 Research Questions

The answer to the first sub-question, What information needs to be communicated to explore an STS?, follows from our task analysis we performed in section 3.2 where we identified that the most common use case for the visualization is during modelling. The user uses the visualization to verify that what they have modelled matches their expectations. To do this they primarily look at the order of the states and transitions. Furthermore the user needs to know to what behaviour a state belongs, the content of transition labels and the transition types.

We used the literature to select the most expressive visual variables to visualize this information. We identified the order of the states and transitions as the most important information. As a result we used the visual variable position to encode this information since that is the most expressive variable. The full visual mapping can be found in section 4.1.1 This answers the sub-question What visual variables are most suitable to encode what aspects of an STS?.

To answer the last two sub-questions, Does semantic zooming improve navigating a visualized STS? and Does semantic zooming help provide the necessary context, while exploring a visualized STS? we evaluated the new visualization method using a questionnaire and a series of experiments. The ques-
tionnaire showed that users were able to navigate a model without losing context and that semantic zooming provided the appropriate context by hiding irrelevant and showing relevant data. This is also confirmed by the experiments, they showed that the users were able to answer questions about the model faster using the new visualization method. The full results of the evaluation can be found in chapter 5.

Using the answers of the four sub-questions we are now able to answer our main research question: Is semantic zooming a more efficient technique compared to the state-of-the-art for exploring symbolic transition systems?

Yes, semantic zooming allows the user to explore a symbolic transition system more efficiently. Were the efficiency of a visualization is determined by how fast users are able to identify and process the information from a model.

7.3 Future work

During this research two directions of future work were identified. Extending functionalities and further evaluation.

During the task analysis three use cases were identified. The new visualization method only supports one of them. Therefore additional functionalities can be researched that will add support for other use cases. For example the visualization can be extended by visualizing test coverage information, which shows what parts of the model are covered by tests. This can also be combined with the semantic zooming where on the model overview level the user should be able to quickly see which behaviours are (partially) covered and once the user zooms in, a more detailed view of what individual states and transitions are covered appears. Something similar can be researched for test traces where a single test trace is visualized.

Furthermore, the questionnaire indicated that the users were unsure if the detail panel is a useful addition. It could be interesting to research if additional transition information is even necessary and if so, how the visualization can provide this information to the user.

Another interesting area to research is how to incorporate parallelism into the visualization. The models created by Axini can contain multiple parallel processes but this was not taken into account while creating this new visualization.

The other direction of future work is that of further evaluating the visualization. Users had no prior experience with the new visualization before performing the user performance experiments. For a more accurate comparison, the users should have roughly the same experience with both visualizations. Furthermore, to generalize the findings in our research a larger sample size is required.
Bibliography


Appendix A

Example of a 3D visualized state transition system

Figure A.1: A state transition system visualized using the interactive 3D visualization method proposed by Ham, Wetering, and Wijk. (A) is the current state, (B) and (C) are possible next states, magenta nodes such as (D) illustrate the visited states. States can also be selected from a list (E) [23]
Appendix B

User performance experiment

B.1 Multiple choice questions

These are all the multiple choice questions used in the user performance experiment. The bold text indicates the correct answer.

<table>
<thead>
<tr>
<th>Example question</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>In ‘as_idle’ a ?SIGNOFF stimulus is performed, which is followed by a !C250_SIGNED_OFF response. This puts the system in what behaviour?</td>
<td>as_idle ss_off as_open</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>In ‘general_errors’ the response !C503_ERROR_STATE is send, a parameterized GET stimulus is performed. This stimulus is followed by a response with two parameters. One of the parameters is ‘CS_EVENT’, what are the two possible values of the other requested parameter?</td>
<td>EV_ERROR OR EV_WARN EV_INFO OR EV_NONE EV_INFO OR EV_ERROR</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>After a succesfull transaction the system is in ‘as_ending’, the state variable ‘total’ is set to 0.0 what response will eventually put the system in ‘as_idle’?</td>
<td>!C260_DATA_PRINTED !C240_TRANS_SUCCEEDED !C233_ACCOUNT_IDLED</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 3</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>The system is in ‘ss_off’ and the ?SIGNON_EXISTS stimulus is performed, a response is send and next a parameterized GET stimulus is performed which will eventually put the system in ‘as_idle’. What is the value of the parameter?</td>
<td>CS_SIGN CS_ACCNT CS_EVENT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 4</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>What behaviour(s) make use of the state variable ‘stored_total’?</td>
<td>as_closed &amp; as_idle as_closed as_open &amp; as_closed</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 5</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>The system is in state ’.105’ in ‘as_open’ there is a constraint on the next transition which uses a state variable, what state variable is used?</td>
<td>stored_total stored_account total</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 6</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Somewhere in ‘main’ the response !C250_SIGNED_OFF is send next the stimulus ?SIGNOFF is performed. What is the name of the state that we are now in?</td>
<td>.49 .40 .28</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 7</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>In ‘main’ the stimulus ?IDLE’ is performed and is responded with !C502_COMMAND_FAILED’. What will be the next stimuli?</td>
<td>?CLOSE ?GET ?SIGNOFF</td>
</tr>
</tbody>
</table>
Question 8
The system is in 'as_ending', a '!C900_TIMEOUT' response is send. What is the name of the state that we are now in?

- 23
- 100
- 103

Question 9
The system is in a state named '_149' which is in 'as_open'. What could be the stimulus that lead to this state?

?ARTREG_NEXIST
?SIGNON_EXIST
?RHCOPY_NEXIST

Question 10
'general_errors' is the only terminating behaviour in this model. How many behaviours only contain one terminating behaviour?

0
1
2

Table B.1: All the multiple choice questions accompanied by the possible answers. The bold answer is the correct one.
## B.2 Results

These are all the results of the user performance experiment. The *visualization* column indicates what visualization was used to answer the question. Where *original* is the visualization that is currently present in TestManager and *new* is our proof-of-concept. The number behind each expert is the number of years experience with MBT that expert has, e.g. expert 1 has 2 years of experience. All the results are the number of seconds it took to answer the question.

### Group A

<table>
<thead>
<tr>
<th>Question</th>
<th>Visualization</th>
<th>Expert 1 (2y)</th>
<th>Expert 2 (3,14y)</th>
<th>Expert 3 (8y)</th>
<th>Expert 4 (2y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Original</td>
<td>78.4</td>
<td>56.8</td>
<td>25.4</td>
<td>59.6</td>
</tr>
<tr>
<td>2</td>
<td>Original</td>
<td>162.0</td>
<td>70.2</td>
<td>51.5</td>
<td>23.8</td>
</tr>
<tr>
<td>3</td>
<td>Original</td>
<td>56.8</td>
<td>61.8</td>
<td>51.2</td>
<td>84.2</td>
</tr>
<tr>
<td>4</td>
<td>Original</td>
<td>53.8</td>
<td>37.1</td>
<td>108.0</td>
<td>58.5</td>
</tr>
<tr>
<td>5</td>
<td>Original</td>
<td>46.1</td>
<td>26.9</td>
<td>21.6</td>
<td>33.6</td>
</tr>
<tr>
<td>6</td>
<td>Original</td>
<td>81.9</td>
<td>61.6</td>
<td>170.9</td>
<td>152.2</td>
</tr>
<tr>
<td>7</td>
<td>New</td>
<td>107.7</td>
<td>96.4</td>
<td>48.5</td>
<td>203.9</td>
</tr>
<tr>
<td>8</td>
<td>New</td>
<td>32.7</td>
<td>67.1</td>
<td>27.4</td>
<td>28.8</td>
</tr>
<tr>
<td>9</td>
<td>New</td>
<td>20.6</td>
<td>34.3</td>
<td>14.5</td>
<td>24.0</td>
</tr>
<tr>
<td>10</td>
<td>New</td>
<td>41.4</td>
<td>11.9</td>
<td>29.4</td>
<td>49.0</td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td></td>
<td>68.1 sec</td>
<td>52.4 sec</td>
<td>54.8 sec</td>
<td>71.8 sec</td>
</tr>
</tbody>
</table>

### Group B

<table>
<thead>
<tr>
<th>Question</th>
<th>Visualization</th>
<th>Expert 5 (7y)</th>
<th>Expert 6 (1,5y)</th>
<th>Expert 7 (15y)</th>
<th>Expert 8 (11y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Original</td>
<td>56.6</td>
<td>26.3</td>
<td>52.9</td>
<td>75.1</td>
</tr>
<tr>
<td>2</td>
<td>Original</td>
<td>68.0</td>
<td>23.4</td>
<td>47.8</td>
<td>30.8</td>
</tr>
<tr>
<td>3</td>
<td>New</td>
<td>83.2</td>
<td>87.9</td>
<td>84.6</td>
<td>97.4</td>
</tr>
<tr>
<td>4</td>
<td>New</td>
<td>13.3</td>
<td>9.1</td>
<td>21.9</td>
<td>8.9</td>
</tr>
<tr>
<td>5</td>
<td>New</td>
<td>74.0</td>
<td>13.3</td>
<td>56.2</td>
<td>28.3</td>
</tr>
<tr>
<td>6</td>
<td>New</td>
<td>23.2</td>
<td>16.1</td>
<td>116.1</td>
<td>19.8</td>
</tr>
<tr>
<td>7</td>
<td>Original</td>
<td>111.0</td>
<td>93.8</td>
<td>82.7</td>
<td>112.1</td>
</tr>
<tr>
<td>8</td>
<td>Original</td>
<td>71.8</td>
<td>37.9</td>
<td>34.0</td>
<td>66.2</td>
</tr>
<tr>
<td>9</td>
<td>Original</td>
<td>62.5</td>
<td>18.1</td>
<td>51.8</td>
<td>27.4</td>
</tr>
<tr>
<td>10</td>
<td>Original</td>
<td>130.2</td>
<td>66.0</td>
<td>71.6</td>
<td>149.2</td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td></td>
<td>69.4 sec</td>
<td>39.2 sec</td>
<td>62.0 sec</td>
<td>61.5 sec</td>
</tr>
</tbody>
</table>
Appendix C

User experience experiment

C.1 Questionnaire results

These are all the results of the questionnaire. After each aspect a number ranging from 1 to 5 is noted where 1 is “Strongly Disagree” and 5 is “Strongly agree”. For the Overall impression aspect the scale was ranging from “Very poor” (1) to “Very good” (5).

<table>
<thead>
<tr>
<th>Expert 1 (2 years MBT experience)</th>
<th>Overall impression</th>
<th>Cognitive complexity</th>
<th>Search Function</th>
<th>Orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Ease-of-use</td>
<td>5</td>
<td>Perceived correctness</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Responsiveness</td>
<td>5</td>
<td>Information relevance</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Perceived efficiency</td>
<td>5</td>
<td>Intuitiveness</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

What are your thoughts on semantic zooming, is it a useful technique for this use case?

Yes. I think it is the quickest way to navigate big and complex models.

What are your thoughts on the search function?

Very useful. Sometimes I found myself searching for the component I was looking for instead of skimming the model to know where to zoom at. One point of improvement: the search results currently go away once I click on something in the list. It would be nice if the results stay there until I dismiss it myself.

Do you have any comments on the visualization in general?

I wish some of my older colleagues would have been still with axini to see such improvement in our visualization. They used to complain a lot about the difficulty of finding elements in the model with the old visualization. I liked both the new visualization and the test to evaluate it.

<table>
<thead>
<tr>
<th>Expert 2 (3,14 years MBT experience)</th>
<th>Overall impression</th>
<th>Cognitive complexity</th>
<th>Search Function</th>
<th>Orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Ease-of-use</td>
<td>5</td>
<td>Perceived correctness</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Responsiveness</td>
<td>5</td>
<td>Information relevance</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Perceived efficiency</td>
<td>4</td>
<td>Intuitiveness</td>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>

What are your thoughts on semantic zooming, is it a useful technique for this use case?

Yes, it shows the exact information to navigate through the model.

What are your thoughts on the search function?

Very helpful.

Do you have any comments on the visualization in general?

I am missing the links to the AML model, but this information is probably not available.
<table>
<thead>
<tr>
<th>Expert 3 (8 years MBT experience)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall impression</td>
</tr>
<tr>
<td>Ease-of-use</td>
</tr>
<tr>
<td>Responsiveness</td>
</tr>
<tr>
<td>Perceived efficiency</td>
</tr>
<tr>
<td>What are you thoughts on semantic zooming, is it a useful technique for this use case?</td>
</tr>
<tr>
<td>What are you thoughts on the search function?</td>
</tr>
<tr>
<td>Do you have any comments on the visualization in general?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expert 4 (2 years MBT experience)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall impression</td>
</tr>
<tr>
<td>Ease-of-use</td>
</tr>
<tr>
<td>Responsiveness</td>
</tr>
<tr>
<td>Perceived efficiency</td>
</tr>
<tr>
<td>What are you thoughts on semantic zooming, is it a useful technique for this use case?</td>
</tr>
<tr>
<td>What are you thoughts on the search function?</td>
</tr>
<tr>
<td>Do you have any comments on the visualization in general?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expert 5 (7 years MBT experience)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall impression</td>
</tr>
<tr>
<td>Ease-of-use</td>
</tr>
<tr>
<td>Responsiveness</td>
</tr>
<tr>
<td>Perceived efficiency</td>
</tr>
<tr>
<td>What are you thoughts on semantic zooming, is it a useful technique for this use case?</td>
</tr>
<tr>
<td>What are you thoughts on the search function?</td>
</tr>
<tr>
<td>Do you have any comments on the visualization in general?</td>
</tr>
</tbody>
</table>
### Expert 6 (1,5 years MBT experience)

<table>
<thead>
<tr>
<th>Overall impression</th>
<th>5</th>
<th>Cognitive complexity</th>
<th>1</th>
<th>Search Function</th>
<th>4</th>
<th>Orientation</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease-of-use</td>
<td>5</td>
<td>Perceived correctness</td>
<td>3</td>
<td>Semantic zooming</td>
<td>4</td>
<td>Interaction</td>
<td>5</td>
</tr>
<tr>
<td>Responsiveness</td>
<td>4</td>
<td>Information relevance</td>
<td>4</td>
<td>Detail panel content</td>
<td>2</td>
<td>World-in-miniature</td>
<td>2</td>
</tr>
<tr>
<td>Perceived efficiency</td>
<td>5</td>
<td>Intuitiveness</td>
<td>4</td>
<td>Detail panel usefulness</td>
<td>2</td>
<td>Encoding</td>
<td>4</td>
</tr>
</tbody>
</table>

What are you thoughts on semantic zooming, is it a useful technique for this use case?

Seems very promising, we should have this.

What are you thoughts on the search function?

Useful for locating specific transitions (for failures or coverage analysis).

Do you have any comments on the visualization in general?

"Very nice, great to use. I only miss the structure / flow between the non-terminating behaviors a bit at the highest level of abstraction."

### Expert 7 (15 years MBT experience)

<table>
<thead>
<tr>
<th>Overall impression</th>
<th>4</th>
<th>Cognitive complexity</th>
<th>3</th>
<th>Search Function</th>
<th>5</th>
<th>Orientation</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease-of-use</td>
<td>4</td>
<td>Perceived correctness</td>
<td>4</td>
<td>Semantic zooming</td>
<td>4</td>
<td>Interaction</td>
<td>4</td>
</tr>
<tr>
<td>Responsiveness</td>
<td>5</td>
<td>Information relevance</td>
<td>4</td>
<td>Detail panel content</td>
<td>3</td>
<td>World-in-miniature</td>
<td>3</td>
</tr>
<tr>
<td>Perceived efficiency</td>
<td>4</td>
<td>Intuitiveness</td>
<td>5</td>
<td>Detail panel usefulness</td>
<td>3</td>
<td>Encoding</td>
<td>4</td>
</tr>
</tbody>
</table>

What are you thoughts on semantic zooming, is it a useful technique for this use case?

Yes I like it. I’m struggling a bit with keeping the overview, but I think that’s better when you know exactly what the model looks like.

What are you thoughts on the search function?

Yes, I think it helps a lot. I like it that I immediately get feedback about the part of the model that the searched item occurs in.

Do you have any comments on the visualization in general?

I like it, much better than our current visualization. I think searching helps a lot. I can imagine that for example explicitly switching off, or abstracting from information that you are not looking for can also help. Maybe in combination with the search functionality. For example you know that the information that you are looking for is on a transition. If you can somehow abstract from states, behaviors etc might speed up the searching.

### Expert 8 (11 years MBT experience)

<table>
<thead>
<tr>
<th>Overall impression</th>
<th>4</th>
<th>Cognitive complexity</th>
<th>5</th>
<th>Search Function</th>
<th>5</th>
<th>Orientation</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease-of-use</td>
<td>5</td>
<td>Perceived correctness</td>
<td>4</td>
<td>Semantic zooming</td>
<td>4</td>
<td>Interaction</td>
<td>4</td>
</tr>
<tr>
<td>Responsiveness</td>
<td>4</td>
<td>Information relevance</td>
<td>4</td>
<td>Detail panel content</td>
<td>3</td>
<td>World-in-miniature</td>
<td>3</td>
</tr>
<tr>
<td>Perceived efficiency</td>
<td>5</td>
<td>Intuitiveness</td>
<td>5</td>
<td>Detail panel usefulness</td>
<td>3</td>
<td>Encoding</td>
<td>3</td>
</tr>
</tbody>
</table>

What are you thoughts on semantic zooming, is it a useful technique for this use case?

certainly

What are you thoughts on the search function?

very helpful

Do you have any comments on the visualization in general?

big step fowards :)