EXPLORING STORY SIMILARITIES USING GRAPH EDIT DISTANCE ALGORITHMS

by

Sritama Paul

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering

Fall 2013

EXPLORING STORY SIMILARITIES USING GRAPH EDIT DISTANCE ALGORITHMS

by

Sritama Paul

Approved: ____
Kristina Winbladh, Ph.D
Professor in charge of thesis on behalf of the Advisory Committee

Approved: ____
Kenneth E. Barner, Ph.D
Chair of the Department of Electrical and Computer Engineering

Approved: ____
Babatunde A. Ogunnaike, Ph.D
Dean of the College of Engineering

Approved: ____
James G. Richards, Ph.D.
Vice Provost for Graduate and Professional Education

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my thesis adviser, Dr. Kristina Winbladh for introducing to the thesis subject and for their continuous support and guidance. Besides my adviser, I would like to thank my fellow research students Sergio Antonio Pino Gallardo and Kwang Choi for providing long fruitful and lively discussions and insightful comments during the whole iMuse project time period. My research would not have been possible without their active participation.

I must acknowledge the excellent encouragement that I received from my distinguished teachers - Dr. Hagit Shatkay and Dr. Mary Sandra Carberry (Department of Computer and Information Science), Dr. Hui Fang and Dr. Chengmo Yang (Department of Electrical and Computer Engineering). I would also like to extend my gratitude to Dr. Stephan Bohacek (Department of Electrical and Computer Engineering) and Dr. Keith Decker (Department of Computer and Information Science) for providing support at the critical juncture of my graduate studies. Also, the excellent support provided by Ms. Kathleen Forwood (ECE department Graduate Coordinator) by patiently guiding me through various forms and procedures during the entire graduate study period, made my life easier as a graduate student.

Last but not the least - I would like to thank the most important people in my life - my parents, my spouse and my daughter. While my parents continuously prayed for my well being, my spouse pushed me towards pursuing higher studies and my little daughter walked along with me in all my hardships.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In computer science, particularly in the fields of interactive storytelling and game authoring, stories are represented as a sequence of goals and actions taken by various characters. Graph data structures are often used to represent these, where the nodes are goals and actions, and the edges represent time and order. Existing story authoring tools allow authors to create stories by adding goals, actions, and characters and thereby create the story. However, existing authoring tools do not provide much help to the author in the form of feedback on the story they are creating. It is therefore difficult for storytellers, particularly novice storytellers, to create their story. One way to aid storytellers, is to have the story authoring tool suggest continuations and details based on some knowledge that the tool has. In our work, we create a knowledge base that exists of all the stories that the tool has collected. The idea is that we can use these existing stories to suggest appropriate feedback to the storyteller. This relies on the assumption that the group of storytellers collectively has knowledge that can be useful to a new storyteller. We know that this is often the case within certain domains such as the military or police force, where many stories or experiences have commonalities. As the stories are represented as graphs, we will use graph similarity algorithms to compute the similarity between the story being authored and the existing stories in the knowledge base. In this thesis, we explore how several different graph similarity algorithms perform with regard to determining story similarity.

Chapter 1

INTRODUCTION

Narratives or story telling has been widely accepted as one of the most effective ways to transfer knowledge and sharing information. Often experienced people share their domain knowledge by narrating an operational situation that they encountered. In other words, the experienced individuals of a domain often tell a ‘story’ about a real situation in which they participated to transfer their domain knowledge to others in the domain. A set of such stories in any domain can produce a valuable information or knowledge base. Generally such stories are expressed in natural language and hence cannot easily be machine processed. However, if the user stories are represented in a structured format, then the information contained in those stories can give rise to a body of actionable knowledge. Actionable knowledge is knowledge that is structured and processed in a way that it can easily be further used to find workflows or patterns in the domain that in turn can be used as a basis for story generation and simulations for example.

From the story authoring perspective, the actionable knowledge base can be used to provide suggestions to authors to formulate their story plot. A new story (which is being written) can be compared with the existing stories in the actionable knowledge base. An intelligent narrative system can suggest a story path at a specific situation for the story being written by analyzing the story paths taken in similar stories in the knowledge base. For this, we need to find effective algorithms to find similarities between the stories.

A story can have a number of characters present in different scenarios and they take different courses of actions in order to achieve their objectives. In other words, a story can have different paths and these story paths can be sequential or parallel. In the field of interactive story and game authoring research, it is a common practice to represent a story as a directed graph, where the nodes and edges are used to represent the sequence of goals and actions of the story characters. In such representations the nodes are goals and actions and the edges describe the order in which goals and actions occur. The circumstances, such as location and other ambient data as well as the actual characters involved in a goal-action sequence are represented as attributes of the edge connecting the goal and action. However, there are different variations of this model – a discussion about different models and our approach of story representation for calculating story similarities are discussed elaborately in later chapters.

Since stories are represented as directed graphs, finding story similarities essentially becomes a study of different graph comparison techniques. In the field of mathematics and computer science, various graph comparison algorithms are available but they are not yet applied in the context of comparing narrative stories. The goal of this study is to apply some of the available graph comparison algorithms to an existing set of stories and evaluate whether results produced by the algorithms are appropriate with regard to story similarity. As evaluation, we will analyze the algorithmic results from a human perspective.

Chapter 2

IMUSE PROJECT

This work is carried out as part of a DARPA funded project: iMuse - **I**nteractive **M**odel Based **U**se Case and **S**torytelling **E**nvironment project [11]. This section briefly introduces the context of iMuse.

The goal of iMuse project is to develop an intelligent authoring tool that will help capturing the experiences of individuals within a domain through their stories, structuring these stories according to a formal model, and making the tool easy to use for non-technical users. The iMuse project also ties into another DARPA program, the Strategic Social Interaction Modules or "Good Stranger" program for which the body of knowledge should consist of the street level encounters between military personnel and people with whom they share little common history, language or culture [11]. The accumulated knowledge from the military personnel can then be used to generate realistic training scenarios for others in the military who lack the ground experience. Since access to military personnel and records of their encounters are difficult to acquire, we used police officers and their experiences as a proxy. The military and police are different in many ways, but they have in common that they perform rigorous scenario based training based on previous experiences and that their experiences are rich in complex social interactions with individuals they share little common history. For this purpose, we have interviewed campus police officers to acquire different street level scenarios related to traffic monitoring and law and order maintenance.

2.1 iMuse Interface

In order to assist the non-technical domain users to formulate their stories, iMuse provides a simple, easy to understand user interface to the storytellers. Using this interface, a storyteller can describe the main events of the story on a timeline and the objective of each action that he/she has taken. A timeline is basically a horizontal line on the iMuse interface placed below the objective field on which the storyteller can add events that have occurred in a chronological order in order to achieve the objective mentioned above the timeline.

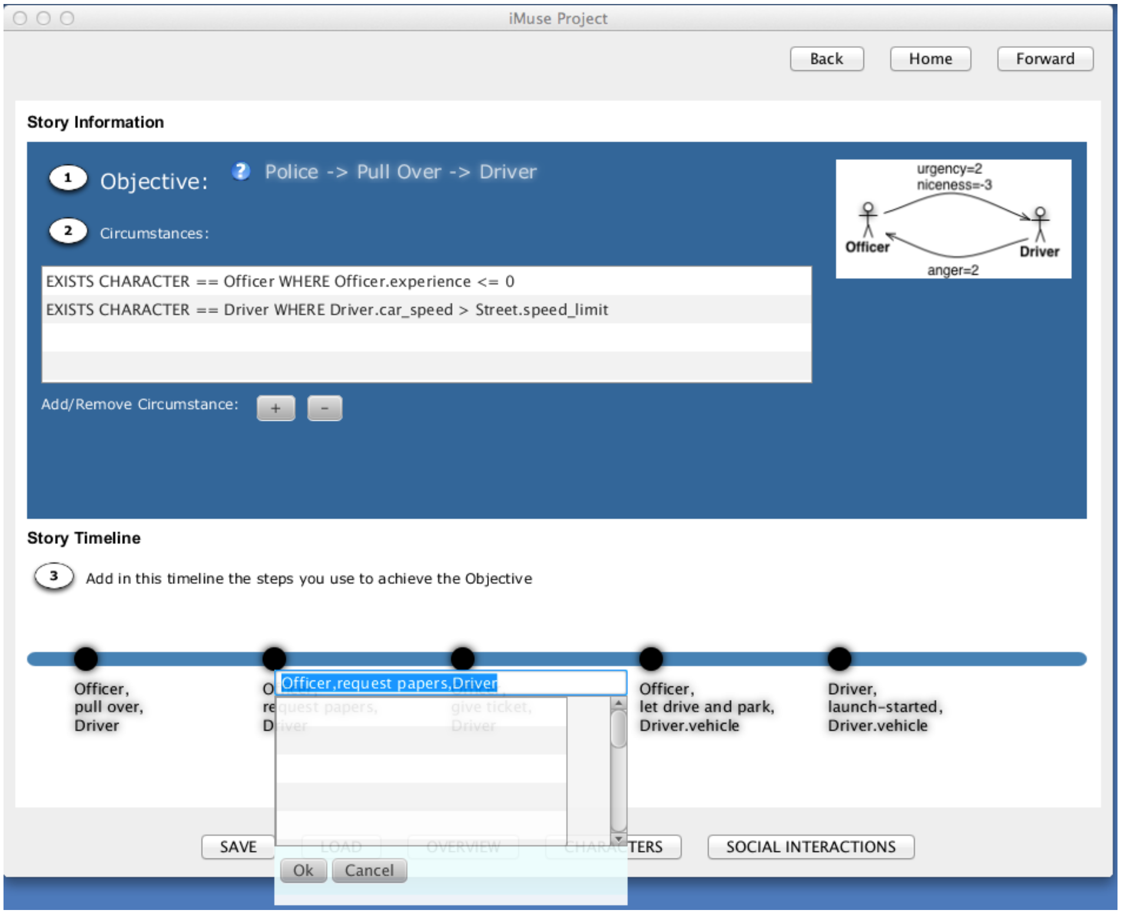


Figure 1 : iMuse Graphical User Interface

In iMuse project context, typically a storyteller will be a police officer or military personnel who will narrate an incident or operational experience that had already occurred as a 'story' using the interface. To narrate the 'story', first, the storyteller needs to define the main objective of the incident or the operation. For example, a traffic police officer can start the narration by entering 'maintain the traffic' as the main objective of his story.

In order to achieve the main objective, the storyteller had to go through a sequence of events during the course of the incident/operation. A good example of such sequence for the above mentioned objective could be as follows: stop a vehicle for over-speeding - > check the driver license - > check for registration -> search the car for illegal arms or drugs etc. - > arrest the driver. All these events can be added in the iMuse interface timeline as steps to achieve the main objective of the operation. Again, each of these events can be further elaborated by mentioning more details that happened during each individual event. To add more details to an individual event say X1, another timeline will be introduced where the storyteller can add another set of the events or actions that happened during main event E1. The individual event X1 (which was created on the first timeline) will contain the objective of the child events that are mentioned in the new timeline. In that way, whenever, the storyteller will want add further details against an event, he can do so by adding a new timeline with a set of the child events.

Using this approach, a user story can be represented as a directed graph, the objective mentioned above the timeline is considered as the goal node of the story graph and the events that are mentioned on the timeline in order to achieve the objective are the action nodes in the story graph. Hence the events or action nodes placed on the timeline can be seen as the child of the user objective or the goal node in the story graph.

As the story events have already occurred, the story author will already know the beginning and the end of the story i.e. the main action nodes of the story can be first entered as the plot outline of story and later further details or granularity can be provided on individual action nodes. To add further consequences for individual events on the time line, a sub story can be developed as the child of that event. In that case, the action node for that event will be transformed into a goal node for which the user can add additional actions on a sub-timeline. A unit distance between a goal node and an action node defines a level in the story.

An example of the story graph structure is given below:

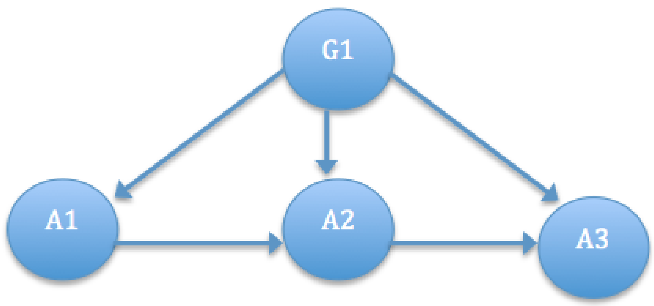


Figure 2 : Sample Story Graph 1

In the above short simple story graph, the main objective is denoted by goal node G1. The action nodes A1, A2, and A3 are the main events that occurred to achieve goal G1. In other words, we can say that A1, A2, and A3 are the main plot outline of the story to meet objective G1. The A1-A2-A3 connection represents the timeline. The directed link between A1, A2, and A3 shows the sequence of actions that is A2 happened after A1 was completed, and A3 happened after A2 was completed.

A sample story corresponding to Figure 2 with meaningful labels is provided as an example. As natural language processing is not part of this thesis, we will not concentrate on the linguistic correctness of the node labels.

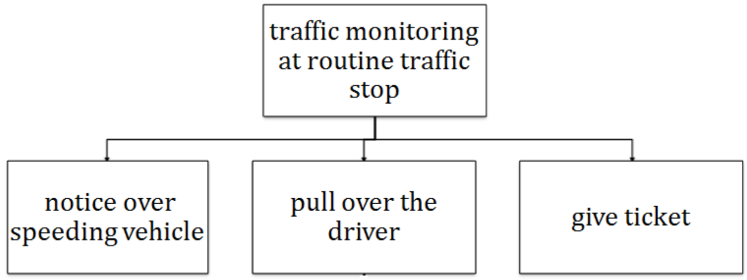


Figure 3 : Story corresponding to Figure 2 graph

Now if the storyteller wishes to add more details to action node A2, another timeline needs to be defined as a child of node A2 that will represent the set of subsequent actions that happened as a consequence of A2. So the set of subsequent actions will have a parent goal node, which is equivalent to A2. For simplicity of representation, we then transform action node A2 as goal node G2 on the main timeline as shown below, which is now the parent of subsequent action node A4. Hence the flow of the story will be G1 → A1 → G2 → A4 → A3.

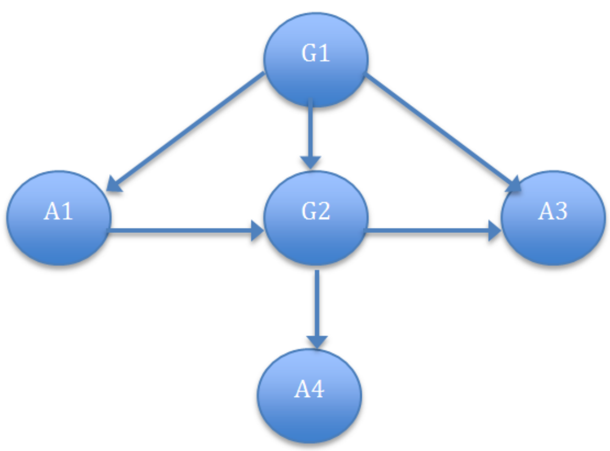


Figure 4 : Sample Story Graph 2

Instead of A2, if the story author decided to add more details to node A1, then A1 would have been substituted with G2. The flow of the story would have been then G1 → G2 → A4 → A2 → A3.

A sample story corresponding to Figure 4 is provided as an example below:

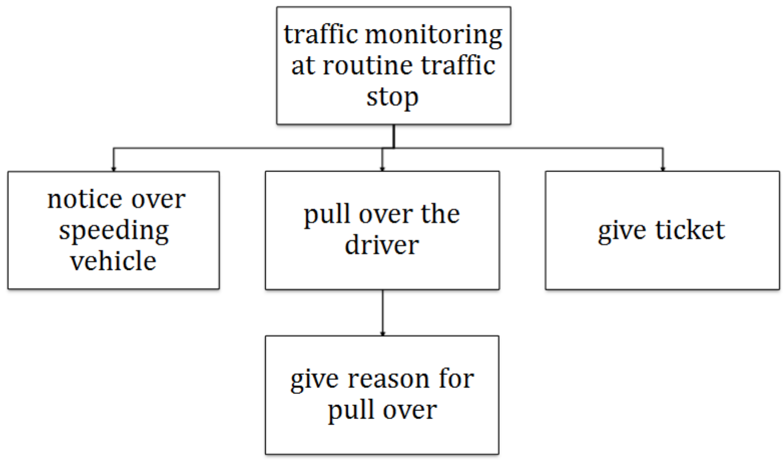


Figure 5 : Story correspondending to Figure 4 graph

In our work, we will maintain that only goal nodes can have a child, hence whenever the story writer adds a subsequent child node to the story graph, the parent node will be then considered as a goal node. Also for the simplicity of the study, we assume that the story graphs will have no repeating nodes.

Chapter 3

RELATED WORK

Two major components of iMuse storytelling tool are the story model, which is used to represent each story in the knowledge base and its easy to use graphical user interface (GUI). In this chapter we present several other story models and interface and discuss their relation to iMuse.

The story authoring models that we studied are used to create interactive games, narratives or used in intelligent tutoring. In the field of multiplayer game authoring, Richard Paul et al. proposed to present the story elements in a hierarchical task network, which can be used to connect in different ways to form a story [1]. A task can then be associated with the characters of the story. Tasks can also be decomposed into smaller tasks. The hierarchy of tasks is represented as a directed tree but the sequence of tasks is not clearly defined. Another story authoring tool that is strongly related to hierarchical task network planning is Wide Ruled [10]. Here the story is represented as a sequence of author goals, plot fragments, and actions which are represented as a directed graph. A plot fragment may have multiple attributes. Plot fragments may include the operational environment attributes, characteristics of the characters involved in the plot and their corresponding attribute/ characteristic values.

Another interesting model of storytelling is the ReQUEST system [3] which was developed by Mark O. Riedl et al. which is based on the QUEST model for question answering for stories [2]. In this model, the story is represented as a directed graph of plot elements, which capture the story events, story states and character goals. The causal relationship between the goal of characters and the narrative events are represented by directed links. There are three types of nodes – event nodes (denotes state changing action in the story), state nodes (denotes snapshot of story situation or circumstance) and goal nodes (goals of the character). Also the relationships between the nodes are of following five types: **Consequence** (such as event 1 causing event 2 to happen), **Reason** (one goal being a sub-goal of another), **Initiate** (event causing a character to take up a goal), **Outcome** (outcome of a goal) and **Implies** (initiating event implying the terminal event node) [4]. The ReQUEST algorithm tries to seek answers to find the reason for taking an action by a story character (the "WHY" questions) and the consequences of an event (the "CONSEQUENCE" questions) [3] by generating questions for hypothetical audience. The human author then answers questions by authoring new plot elements. If the questions are not answered, it results in disjointed story graphs [4].

In the same concept of using question and answering for generating a more complete story, the AQUA authoring model was proposed by Ashwin Ram [5]. In AQUA, when a new story is fed to the system, AQUA tries to answer a set of existing questions stored in memory. In order to answer the questions, more details are added to the story. The enriched information is then used to generate more refined questions to the story that is being written. The construction of the story elements and its explanation of the question can be represented in a tree structure. Like ReQUEST, this model also uses different types of nodes and causal links for explanation. AQUA mostly focuses on finding explanations of concepts learned from the story and refining the story in order to provide a valid explanation to the questions that are generated by the system.

Authoring tools like Scribe also uses preconditions, actions and post-condition nodes and the plot points are connected in graph structure [6]. However, there is another authoring tool named Cyranus [7], which employs a hierarchical transition network with rule-based system. The hierarchical transition network is derived from Harel’s state charts [8].

In all the story authoring model studied, most of them do not provide any scope of knowledge base creation from saved stories and provide aid in formulating a new story from the information stored in the knowledge base. Apart from the ReQUEST model [3], other models do not provide any notions of suggesting forward story paths. The ReQUEST model plans to produce inferences by using Superordinate Goals, Causal Antecedents and Causal Consequences [9]. However, we notice that most of the research related to providing suggestion in ReQUEST is still under progress.

From the above study of various story models, it is clear that representing a story, as a directed graph is the most widely accepted representation. The essential parts of a story are goals of the characters, pre-conditions (although they can be differently termed in different models) and actions. However, different models use different types of graph representation. Some of the models represent preconditions or characteristics of the situation as a node in the story graph whereas some other models represent the characteristics of the situation as the attribute of the link between a goal and corresponding action. As we study different story models, we find that a plot fragment or precondition may consist of various elements like story environment, character traits, degree of severity of certain conditions etc. Hence a precondition can be a composite entity and will be difficult to represent in a story graph. So for the story graphs that we consider for this thesis will include only goal and action nodes. Besides, in many cases, suggesting future story paths may not be heavily dependent on the circumstances of the story situation. For example, some routine investigation steps that are regularly followed by police officers will not depend on story circumstances involving traffic control. Hence we exclude preconditions as nodes in the story graphs used for our study. Rather we would like to keep them as attribute to the relationship/link between the goal and actions nodes although the pre-conditions will not be used for comparing different story graphs. However, including precondition for the story graph comparison will remain a future research criterion.

Chapter 4

STORY COMPARISON

In the previous chapters, we have discussed that in iMuse project, user stories will be represented as directed graphs. In order to find similarity between two stories, we need to compare the underlying graph structures. Hence to find the similarities between two stories, we need to employ some graph comparison algorithms. The goal of the current study will be to find suitability of the existing graph matching algorithms for finding similarities between two story graphs. However, before applying the graph matching algorithm on the stories, it is essential to define the criteria for similarity comparison between two graphs to determine the effectiveness of the graph comparison algorithms for this task.

4.1 Story Comparison In This Study

There are many ways the similarity between two stories can be defined. But a precise way of defining story similarities has not been established previously. Similarity between two stories may depend on the context or domain of the stories as well as the story authoring model. Even in a specific story authoring model, the similarity between stories can depend on a number of factors such as number of common events, sequence of events etc.

When user stories are represented as directed graphs, two stories can be similar if they have a large number of common nodes. Despite having a large number of common nodes, the stories can still be different if the nodes appear in different locations in the two stories. For example, let us compare two story graphs $Sg1$ and $Sg2$ having a number of common nodes and having the same graph structure. The root node in a story graph will represent the main objective of the story. However, in story graph $Sg1$, the common nodes are present in its lower levels where as in story graph $Sg2$, those common nodes are present in the higher level (the level of a node in a story graph can be considered as inversely proportional to the minimum number of nodes required to be traversed from the root node to its location).

Now in iMuse context, two stories having a number of common nodes indicates that the stories share a number of common events or objectives that resulted in those events. Since for story graph $Sg2$, the common nodes are in the higher level compared to story graph $Sg1$, the comparison between two stories will indicate that story graph $Sg2$ has more details about the follow up events resulting from common events. On the other hand, it can be said that the story graph $Sg1$ vastly differ in its objectives that resulted in the common events with story graph $Sg2$. Hence, we can infer that the degree of similarity between two stories will depend not only on the common nodes but also how they are connected in the stories. In other words, we can say that two most prominent factors in finding similarity between story graphs are the number of common nodes and number of common edges.

In addition to the above two factors, while comparing two stories, it also needs to be considered whether to give equal weightage to the number of common nodes and number of common edges. Another factor in finding the similarity between the story graphs could be whether the weightage of the common nodes will vary depending on the distance between the root node and the common nodes.

Another situation is that the whole of a story graph $Sg1$ is present as a sub graph of another story graph $Sg2$. If the sub graph constitutes a small part of the $Sg2$, the similarity score may be less if there is another graph with higher number of matching nodes and edges (but is not a sub graph of the $Sg2$). Similarly, the location of the sub graph (sub graph closer to the root node of the graph compared to sub graph present in a lower level) in the story graph may also influence the similarity measure between two graphs.

In iMuse, the nodes of the story graphs are identified by their text labels. Different story authors may describe the same events using different textual representation. Since linguistic comparison of texts is out of scope of our study, we will rather focus on the graph structure of the stories to find similarity between them rather than comparing the node labels. Also, different authors can describe the same incident using different story structures. In that case, defining the story similarity between the stories can be very much subjective even if the story structures are very much dissimilar.

From the above discussion, it can be inferred that a number of factors can influence the similarity function between story graphs, which can be extremely complex and be subjective to human judgment. We have not found any well-known comparative study on the various story similarity criteria so far. However, many researchers define the similarity between two graphs by finding the number of steps required to transform one graph to another. The technique to transform one graph to another is generally referred as graph edit distance algorithms.

From the iMuse design context, we also noticed that finding story similarities based on graph edit distance algorithms can provide a good preliminary research ground as this technique considers comparison of directed graphs as well as the number of common nodes and edges between the compared graphs. Most of the story comparison criteria that we found to be relevant in comparing stories are included in graph edit distance based algorithms.

In this study, we wanted to find whether the different graph edit distance based algorithms are consistent in finding the similarities between the story graphs as well as whether the similarity measures found by the different algorithms are aligned with the judgment of similarity by a human reader of the stories.

The below section will introduce a brief introduction of few graph edit distance algorithms used in our study.

4.1.1 Graph Edit Distance Algorithms

A graph edit distance algorithms defines the minimum number of editing operations (such as insertion, deletion, substitution of nodes and edges) that are required to transform one graph to another. If the graphs are very much dissimilar, a large number of editing operations will be needed i.e. one graph needs to be strongly modified to be transformed into the other. Often a cost factor is associated with each of the editing operations. Hence two graphs will be most similar if they have the minimum cost edit paths. Formally the minimum cost edit distance between two graphs g_1 and g_2 is defined by [14]:

$$d(g_1, g_2) = \min_{(e_1,\ldots,e_k)\in\Upsilon(g_1,g_2)} \sum_{i=1}^{k} c(e_i)$$

Figure 6 : Minimum Cost Edit Distance

In the above notation, $e_1, \ldots e_k$ denote the k number of edit operation required to transform graph g_1 into graph g_2, $c(e_i)$ denotes the cost of each edit operation, and $\Upsilon(g1, g2)$ denotes the set of set of all possible edit paths between two graphs g_1 and g_2 [14].

In this study, we considered the following graph edit distance based algorithms:

- A* Algorithm
- Beam Search
- Bipartite Graph Edit Distance using Hungarian Algorithm
- Bipartite Graph Edit Distance using Volgenant and Jonker Assignment Algorithm

A brief description of the algorithms is noted below:

A-Algorithm (exact graph edit distance)* – A* algorithm is a widely used algorithm for calculating exact graph edit distance. It is basically an ordered tree search algorithm where the search starts from the root node of the tree. The search tree is constructed dynamically by iteratively creating successor nodes linked by edges to the currently considered node [14], [15]. Formally, if for a node p in the search tree, $g(p)$ denotes the cost of the optimal path from the root node to the current node p, and $h(p)$ denotes the estimated cost from p to a leaf node, the sum $g(p) + h(p)$ gives the total cost assigned to an open node in the search tree[14]. This algorithm is guaranteed to find an optimal edit path [15]. However, we have not used this algorithm in our experiments due to resource and time constraint.

Beam Search [16] – Since the computationally complexity of A* algorithm is exponential in the number of nodes of the involved graphs, the running time and space complexity may be huge even for reasonably small graphs. So often simple variations of the standard A* algorithm is used for calculating graph edit distance between two graphs. In case of beam search, only a fixed number of s nodes are used as a set of open nodes. So for a new partial edit path is added, only the s partial edit paths p with the lowest costs $g(p) + h(p)$ are kept, and the remaining partial edit paths are removed. Hence the full search space is not considered but since the partial edit paths with lowest costs are considered, a nearly optimal edit path is obtained [14].

Bipartite Graph Edit Distance using Assignment Algorithms (Hungarian and Volgenant and Jonker) [17], [18], [19], [20], [21] – Another variation of computing graph edit distance between two graphs in a faster way is to compute a cost matrix for transforming one graph to another. Bipartite optimization procedures applied to the cost matrix, which corresponds to the minimum cost mapping of the nodes and their local edge structure of the compared graphs. This procedure produces suboptimal graph edit distance between two graphs, which is equal to or greater than the exact edit distance. Hungarian Algorithm and algorithm of Volgenant and Jonker are two well known bipartite optimization procedures which can be used in cost assignments.

4.2 Study Design

In this study, we have designated one story graph (denoted as U or the base graph) as the story that will be compared with a set of story graphs. Each of the algorithms (except A* algorithm) mentioned in the previous section will compute a similarity score for a set of story graphs against the base story graph U. The stories will be then ranked in the order of similarity score with respect to the base graph on the basis of the results obtained from each algorithm. The ranks obtained from each algorithm will be then compared with the ranks obtained from other algorithms. Finally the ranks obtained from the algorithms will be compared with the ranks obtained from human readers. Details of the ranking provided the human readers are noted in section 4.5.

We have created a set of 14 smaller story graphs in this study to compare with U, in which the number of uncommon nodes with respect to U is very less. From the story authoring point of view, some of the smaller stories may just show the basic structure of the story outline. All the stories are based on traffic monitoring situations faced by the police officers. [1]

In the set of stories, different story graphs are created by changing the positions of the nodes or adding or deleting more peer and child nodes from the basic story graph U. The different node organization in the stories depicting the same operational scenario implies that different story tellers will organize their narrations in different manners. Comparing those story graphs with the basic story graph U will indicate how much the story similarity measure obtained from different algorithms will be affected by small modifications of the node position and change of node numbers. The similarity measures obtained from different algorithms can also indicate about the factors influencing the story similarity in a particular algorithm and provide

[1] The domain of iMuse is stories related to military operations but since access to military personnel is highly restricted, we relied on the similar domain stories from police department.

some pointers about different parameter optimizations while using the algorithm for comparing stories.

We apply the findings of the experimental results on the smaller stories with the observation obtained from human readers of the stories. The results from the above study will form the base of further research about applying the graph edit distance algorithms in the context of story comparison. Different researchers may then apply the findings to define their respective story similarity functions.

4.3 Story Representation

A story graph can be represented in various ways – one of the ways is to draw the graph i.e. visually represent the graph, but we cannot use that representation as input to any of the tools. Another way is to represent the graph in XML format. An XML format can maintain the hierarchical structure of the story graph. Moreover, using the XML format, we can also store the various attributes of the nodes and the connections, which is essential in representing the story characteristics in various story authoring models. In fact, a graph exchange language (GXL) is introduced for standard representation of graphs in XML format [12]. An example of the sample story graph 1 (refer to Figure 1) represented in GXL format is shown below:

```
▼<gxl>
  ▼<graph id="U" edgeids="false" edgemode="directed">
    ▼<node id="_1">
      ▼<attr name="label">
          <string>G1</string>
        </attr>
      ▼<attr name="type">
          <string>goal</string>
        </attr>
      </node>
    ▼<node id="_2">
      ▼<attr name="label">
          <string>A1</string>
        </attr>
      ▼<attr name="type">
          <string>action</string>
        </attr>
      </node>
    ▼<node id="_3">
      ▼<attr name="label">
          <string>A2</string>
        </attr>
      ▼<attr name="type">
          <string>action</string>
        </attr>
      </node>
    ▼<node id="_4">
      ▼<attr name="label">
          <string>A3</string>
        </attr>
      ▼<attr name="type">
          <string>action</string>
        </attr>
      </node>
      <edge from="_1" to="_2"/>
      <edge from="_1" to="_3"/>
      <edge from="_1" to="_4"/>
      <edge from="_2" to="_3"/>
      <edge from="_3" to="_4"/>
    </graph>
  </gxl>
```

Figure 7 : Sample Graph 1 in GXL format

In our study, we will use all the stories in standard GXL format before giving as an input to the algorithms. As shown in the above notation in Figure 7, each node is identified by a unique string, which is of "_x" format where x is an integer number. We define two attributes for a node – which are label and type. The label attribute of a node will represent the corresponding node label and the type attribute of the node will indicate whether the node is a goal node (representing an objective) or an action (representing an event) node. If the node is a goal node (i.e. the node can have a child action node), the node type attribute will be 1. Otherwise, if the node is an action node (i.e. the node cannot have a child node), the node type attribute will be 2. For edges, there is a distance attribute defined whose value is set to 1 for all the edges.

The following figure shows the visual representation of the basic story graph U that is to be compared with the set of smaller and bigger stories:

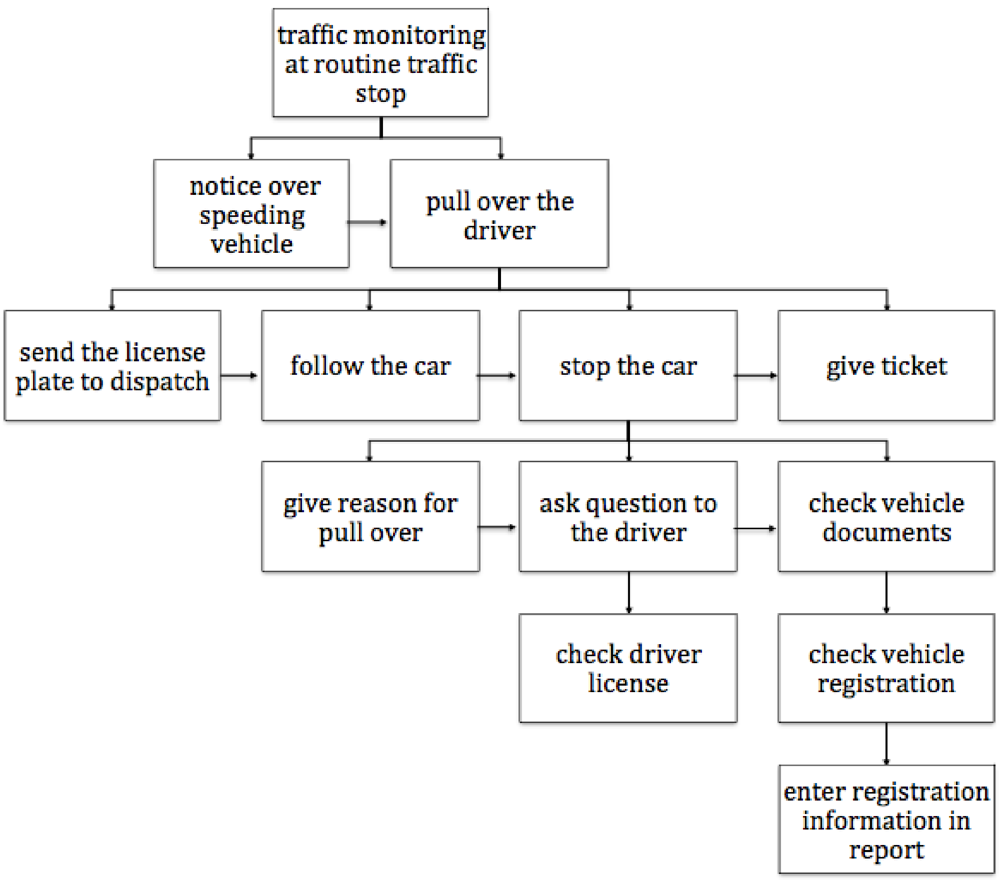


Figure 8 : Basic Story Graph U

The structures of the other individual stories are provided in the Appendix A and B in GXL formats as well as in visual representations. The set of stories used in the study are constructed using the below nodes. The stories will share some common nodes with each other; as well as some other nodes. The below table contains the list of nodes that are used in formulating the sample stories.

Node Identification Number	Node Label Name
1.	traffic monitoring at routine traffic stop
2.	notice over speeding vehicle
3.	pull over the driver
4.	send the license plate to dispatch
5.	follow the car
6.	stop the car
7.	give ticket
8.	give reason for pull over
9.	ask question to the driver
10.	check vehicle documents
11.	check the traffic condition
12.	allow the driver to drive till next parking place
13.	check driver license
14.	check vehicle registration
15.	check vehicle insurance status
16.	find traffic is blocked
17.	find license is expired
18.	enter license information in report
19.	find registration is ok

Node Identification Number	Node Label Name
20.	enter registration information in report
21.	find insurance is ok
22.	enter insurance information in report
23.	suspend driver license
24.	check influence of alcohol
25.	take sobriety test of driver
26.	driver fails in sobriety test
27.	find license is ok
28.	check possession of illegal arms
29.	check possession of drugs
30.	find no arms
31.	find no drugs
32.	arrest the driver
33.	capture illegal arms
34.	maintaining security near oil field
35.	notice unauthorized vehicle approaching
36.	bring out the weapon and take position
37.	search the vehicle
38.	let the driver go
39.	call for backup security
40.	approach car from passenger side
41.	ask for reason to enter premises
42.	check possession of explosives

Node Identification Number	Node Label Name
43.	find that driver lost his way
44.	find no explosives
45.	driver does not have vehicle registration documents
46.	maintain law and order of a city
47.	patrolling the roads
48.	submitted report to the supervisor
49.	check previous violation records
50.	maintain traffic in the highway
51.	ordered the driver to stop through loudspeaker
52.	check if speed limits are maintained
53.	clear rest of the traffic
54.	check license of arms
55.	check if drivers following traffic signals

Table 1: List of story nodes used in the story graph

4.4 Characteristics Of The Sample Stories

The following table briefly discusses about the characteristics of the set of story graphs used in the experiment and how they differ with respect to the basic story graph U (as shown in Figure 7). For more details about the story graphs, please visit the appendix section.

Story Identifier	No. Of Common Nodes	No. of Different Nodes in the Story Graph + No. of Different Nodes in U	No. Of Common Edges	No. Of Different Edges in the Story Graph + No. of Different Edges in U	Characteristics
E1	10	5 + 3 = 8	15	6 + 3 = 9	The end nodes i.e. the lowest level nodes of E1 differ from the basic story graph U. There are four new additional nodes below 'ask question to the driver' node and one different node below 'check document status' node in E1.
E2	13	3 + 0 =3	18	5 + 0 = 5	The basic story graph U is a sub-graph of story graph E2; there is one extra small sub-graph under 'check vehicle documents' node in E2.
E3	12	3 + 0 = 3	18	5 + 0 = 5	The basic story graph U is a sub-graph of story graph E3; but there are some extra story nodes at the top level of the sub graph E3 i.e. an extra sub-graph is there in E3 above the root level of basic graph U.
E4	13	1 + 0 = 1	17	3 + 1 = 4	There is only one extra node 'check

Story Identifier	No. Of Common Nodes	No. of Different Nodes in the Story Graph + No. of Different Nodes in U	No. Of Common Edges	No. Of Different Edges in the Story Graph + No. of Different Edges in U	Characteristics
					previous violation records' in the story graph E4 compared to the basic story graph U. The extra node is placed in the mid-level of the story graph in E4.
E5	12	1 + 1 =2	16	2	Story graph E5 is identical to basic story graph U – except that the root node is different.
E6	12	1 + 1 = 2	15	6 + 6 = 12	One node in the basic story graph ('stop the car') is replaced with another node ('ordered the driver to stop through loudspeaker') in story graph E6. The node that was replaced in E6 was in the mid-level of the story graph E6.
E7	13	2 + 0 = 2	17	3 + 1 = 4	Story graph E7 has two additional nodes ('take sobriety test of driver' and 'driver fails in sobriety test') between two consecutive nodes ('ask question to

Story Identifier	No. Of Common Nodes	No. of Different Nodes in the Story Graph + No. of Different Nodes in U	No. Of Common Edges	No. Of Different Edges in the Story Graph + No. of Different Edges in U	Characteristics
					the driver' and 'check driver license' in the lower levels of basic story graph U.
E8	12	1 + 0 = 1	11	8 + 7 = 15	Story graph E8 has one additional node inserted between 'pull over the driver' and 'stop the car' nodes in the basic story graph U – thus the 'stop the car' node and subsequent child nodes are pushed below one level in E8.
E9	13	2 + 0 = 2	16	5 + 2 = 7	The story graph E9 has two additional nodes ('check if drivers following traffic signs' and 'check if speed limits are maintained') just below the root node of the basic story graph U – thus creating one additional level in the top in story graph E9.
E10	13	0 + 0 = 0	11	11 + 11 =	The base story graph U and E10

Story Identifier	No. Of Common Nodes	No. of Different Nodes in the Story Graph + No. of Different Nodes in U	No. Of Common Edges	No. Of Different Edges in the Story Graph + No. of Different Edges in U	Characteristics
				22	has all nodes common, only the positions of two common nodes ('stop the car' and 'pull over the driver') are interchanged.
E11	13	1 + 0 = 1	18	1 + 0 = 1	E11 has one additional node ('maintain law and order of a city') on top of the root node of the basic story graph U – thus changing the root node of the two stories.
E12	7	1 + 6 = 7	9	3 + 9 = 12	Story graph E12 does not have much details at the lower level of the story, hence the number of levels are less than the base story graph U.
E13	7	1 + 6 = 7	10	2+ 9 = 11	E13 is almost identical with E12 except the position of one additional node ('check traffic condition'), which is moved to the end of the timeline in the same level.

Story Identifier	No. Of Common Nodes	No. of Different Nodes in the Story Graph + No. of Different Nodes in U	No. Of Common Edges	No. Of Different Edges in the Story Graph + No. of Different Edges in U	Characteristics
E14	12	1 + 0 = 1	12	8 + 3 = 11	E14 story graph has one additional node (‘clear rest of the traffic’) than the base story graph U but two other node positions (‘give ticket’ and ‘stop the car’) are changed in such a way that it created a number of different edges.

Table 2 : Characteristics of smaller story graph songs

4.5 Human Reader Ranking

To validate the similarity ranking produced by the graph edit distance algorithms, we need to compare the similarity ranking results with the similarity ranking given by human readers of the stories. If the ranking produced by an algorithm under certain parameters matches with the ranking provided by the human readers, then we can say that algorithm is suitable for finding story similarities in iMuse context.

We conducted a user study with a small group of people – each person in the group are given the graphical representation of the user stories (as shown in Appendix A) and the picture of the basic story graph U. They are also given a bunch of small rectangular flash cards – each of the cards had a node label printed on them. The human readers were asked to use the cards to construct the stories. Each reader is then asked to rank the user stories based on their similarity with basic story graph U. The stories that were most similar with the base story should be in the higher order of the ranking and the stories that were most dissimilar with the base story should be in the lower order of the similarity ranking.

Since, the human judgment is subjective, we also provided the following guidelines to the human readers to judge similarity the stories and the basic graph U:

- Compare the number of common nodes and number of uncommon nodes between the basic story graph U and the other story graph that is to be compared with U.
- Compare the edges between the basic story graph U and the other story graph that is compared with U.
- Compare if the story graph to be compared has the same structure as the basic story graph i.e. whether for each node and edge in basic story graph U, there is a corresponding node and edge in the compared user story graph (commonality of structure).
- If the basic story graph needs to be converted into the other user story graph, how many steps would you require? (The human readers were instructed to use flash cards with the node labels at this point).
- Do you consider whether changes in the main story plot points have more weightage than the changes in the story details when you compare two stories? In other words do you consider, the distances between the root node and the nodes that are replaced to convert a user story to the basic story are important or not?

The rankings obtained from the group of human readers are not unanimous. We have observed the following facts from the ranking obtained from the human readers:

a. All human readers thought that story graphs E12 and E13 are the most dissimilar story graphs compared to basic story graph U
b. All human readers agreed that story graphs E4, E5, E6, E10 and E11 are the story graphs which are most similar with the basic story graphs (although the relative ranking among these five story graphs differ for different human readers)
c. Story graph E2, E7 have higher ranking than story graph E1
d. Story graph E9 have higher ranking than story graph E3
e. Story graph E11 have higher ranking than story graph E9

We also noticed that some of the human readers thought that adding a new node to a new position in the basic user story (in order to transform the basic user story to the comparing user story) causes more structural difference with the basic story than interchanging the positions of nodes. Those readers place story graph E10 in a higher ranking than story graph E11.

Some readers, who used flash cards to transform the basic story to the other story, also thought that interchanging node positions to transform one story graph to another has more steps than adding a new node to the story. Also, according to those readers interchanging node positions causes greater changes to the edges of the story graphs – hence even though the structure of the two compared story graphs remain same but the sequence of the story actions will greatly differ. According to those readers, story graph E11 is more similar to basic story graph than story graph E10.

Some readers gave more emphasis on having more common structure with the basic graph in higher level compared to the lower level. As per their opinion, changes in the higher level of the story have the potential to change the entire story plot line. They argued that the effect of change in the higher level of the story (i.e. the effect of changes in the story plot) is likely to cause the changes in the lower level of the stories also since the change in story plot would also cause the changes in the course of actions as the story progresses.

Due to the above reasons, the ranking of the stories from different human readers differ specially in the middle order of the ranking. But since all the readers were in agreement for the top level and bottom level ranking of the stories, we accept observation (a) and (b) as the main filtering criteria for the results obtained from different algorithms under different parameter setting i.e. if any algorithm under specific parameter setting ranks the story graphs in such a way that contradicts observation (a) and (b), that result is considered to be invalid.

4.6 Graph Matching Toolkit

For our study, we used a graph matching toolkit software developed by K. Riesen and others at Institut für Wirtschaftsinformatik [14]. We used Beam search, Bipartite Graph Edit Distance using Hungarian algorithm and Bipartite Graph Edit Distance using Volgenant-Jonker algorithm for computing similarity score between the stories.

The graph matching toolkit is developed using Java programming language and provides a GUI where a number of parameters can be set for graph comparison.

The below figure displays the main UI (partial view available) of novel graph matching toolkit used for this study.

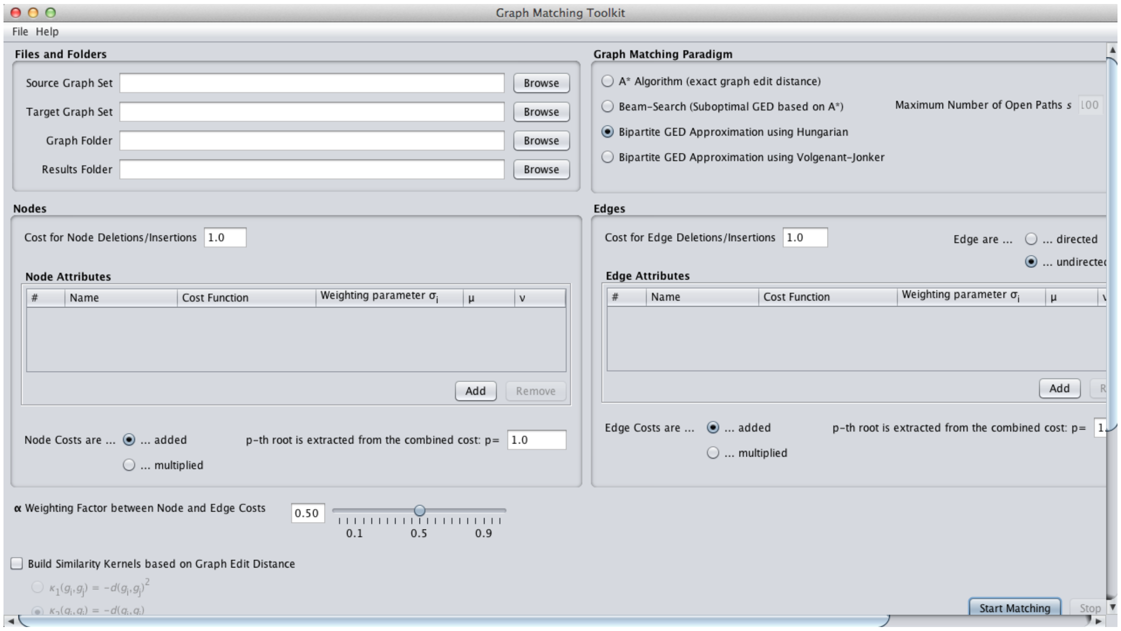


Figure 9 : The main UI of Graph Matching Toolkit

As seen from the above diagram, one can choose different algorithm for graph comparison as well as they can define whether the graph edges are directed or undirected. We can also define different attributes for a node or for an edge, which are to be considered for computing graph edit distance (in the gxl file, we may provide more attributes for a node or edge, but all the attributes need not be considered for computing graph edit distance). For each of the defined attributes, a weighing parameter σ_i can be specified which will define the importance of the attribute for the node or edge. To calculate the cost of individual edit operations of a node or edge, first the distances between the individual attributes are computed and multiplied by the weighing parameter σ_i of the individual attributes. Then we can either add up or multiply (which can be set in the GUI) the individual weighted attribute distance values to get the individual node or edge editing distance values, which is then multiplied by individual node or edge editing cost. In that way, the cost of editing all the nodes (say $\sum N$) and the cost of editing all the edges say (say $\sum E$) is computed. Further we can define, a weighing factor $0 \leq \alpha \leq 1$ which decides whether the edit operation cost on the nodes or on the edges is more important. The final graph editing distance between two graphs is defined as $\alpha \sum N + (1- \alpha) \sum E$. The default setting is $\alpha = 0.5$ leading to balanced importance between node and edge operation cost [14].

4.6.1 Input And Output Of The Tool

The individual graphs that are to be compared with each other are to be saved in .gxl files (represented in graph exchange language format) and placed in a folder. If there are n story graphs that are to be compared with each other, the tool will generate a comparison matrix of $n\ X\ n$ dimension. The value in the i th row and j th column will indicate the weighted graph edit distance value of converting i th graph into j th graph. The graph with the lowest weighted edit distance with respect to U will have highest similarity with it.

The different combinations of input parameters and algorithms that are used for experiments are shown in the below table.

Algorithm Used	Cost of Edit Operations for Nodes and Edges	Number of Node Attributes Used	Number of Edge Attributes Used	Addition/ Multiplication of Node Attribute Costs	Addition/ Multiplication of Edge Attribute Costs	Weighting factor α
A*	Cost of edit operations of nodes and edges are equal	0	0	Add	Add	0.5
Beam Search	Cost of edit operations of nodes is higher than that of edges	1 (node type)	1 (optional distance attribute)	Multiply	Multiply	0.75
Bipartite GED Approximation using Hungarian Algorithm	Cost of edit operations of nodes is lower than that of edges					0.25
Bipartite GED Approximation using Hungarian Algorithm						

Figure 10 : Different Parameter and Algorithm Combinations of Graph Matching Toolkit

The above combination can give rise to a large number of experiments. Hence we optimize some of the combinations for our experiments. As explained earlier, we do not consider A* algorithm to avoid computational and resource complexity of larger graphs. Besides, we will only use addition of node and edge attribute costs. This will reduce the number of combinations to 3 X 3 X 2 X 2 X 3 = 108. The similarity score between two graphs is computed as the squared graph edit distance between them – the graphs having minimum graph edit distances will be most similar.

The experimental results obtained from the above combinations are presented and analyzed in the next chapter.

Chapter 5

EXPERIMENTAL RESULTS USING GRAPH EDIT DISTANCE ALGORITHMS

In our experiment, the graph matching toolkit compares the basic graph U with 14 other smaller graphs ($E1..E14$) and the story graphs are sorted on the basis of lowest squared graph edit distance between the graph and U.

We start our experiments with the simplest parameter setting[2] – that is we apply individual algorithms with considering no node or edge attributes (i.e. node type attribute value of a node is not considered for computing the similarity score between the graphs) of the graphs and equal cost for individual node and edge edit operation. We vary the importance of node and edge editing operations by changing the value of α in the Graph Matching Toolkit. Greater the value of α, greater will be the importance of node editing operation. Below table shows the result from different algorithms with the above parameter settings. The story graphs placed in the same row have the equal similarity score.

---

[2] Before going through the experimental results and the analysis, it is recommended to go through the story graph structures in Appendix A. The results and analysis are made with the assumption that the reader is familiar with the story structures.

Experiment 1:

Algorithm Used	Cost of Node Insertion/ Deletion	Cost of edge Insertion/Deletion	Number of Node attributes used	Number of edge attributes used	Weighting factor α = 0.5	Weighting factor α = 0.75	Weighting factor α = 0.25
Beam Search	1	1	0	0	E5, E6 E10 E11, E14, E4 E8 E1, E7, E9 E3 E12, E13,E2	E5, E6 E10 E11, E14, E4 E8 E1, E7, E9 E3 E2 E12, E13	E5, E6 E4, E11, E14 E8 E1, E7, E9, E10, E12, E13 E3 E2
Hungarian	1	1	0	0	E5, E6 E4, E8 E7 E1 E2 E10, E14 E9, E11 E12, E13 E3	E5, E6 E4, E8 E10 E7 E1 E14 E2 E11 E9 E3 E12, E13	E5, E6 E4, E8 E7 E1 E2 E14 E10 E12,E13 E9 E3 E11
VJ algorithm	1	1	0	0	E10 E4, E5, E6 E8 E12, E13 E11 E7, E14 E1 E9 E3 E2	E10 E5, E6 E4 E8 E11 E14 E7 E1 E9 E3 E2 E12, E13	E12, E13 E10 E4 E5, E6 E8 E7 E11 E1 E14 E9 E3 E2

Table 3 : Result with no node and edge attributes and equal cost for individual node and edge operation

Beam Search Algorithm: In the result shown in the table above, it is observed that for α = 0.5 or 0.75, the similarity ranking of graphs is almost same when Beam Search algorithm is applied with the above parameter settings. In all three values of α, as per Beam Search algorithm, the story graphs E5 and E6 have highest similarity with basic story graph U and story graphs E3, E2, E12 and E13 have lower similarity with the basic graph.

One important observation from the result is that story graph E10 have lower similarity when α = 0.25 i.e. edge editing operations have more importance than node editing operation but have higher similarity when α = 0.5 or 0.75 i.e. node editing operation has equal or higher importance than edge editing operation. In E10 graph, only the position of two nodes is interchanged i.e. U and E10 have all the nodes in common but since the position of two nodes are interchanged, the edges are changed. Since with α = 0.25, the weightage of edge edit operation is more than node edit operation, the story graph E10 is penalized heavily.

Another important observation is that relative lower ranking of story graph E11 than story graph E5 and E6. E11 has one additional node at the top of the basic story graph – whereas E5 and E6 maintain the exact story structure. Hence it can be inferred that the structural similarity of the graphs impact the similarity ranking to a greater extent compared to the number of common and different nodes and edges using this algorithm. This can be further supported by the fact that story graphs E6, E14 and E8 rank higher due to higher structural similarity with U despite having large number different edges.

Story graphs E2, E3, E12 and E13 are ranked lower in the similarity using this algorithm. Basic story graph U is a sub graph of E2 and E3 and story graphs E12 and E13 are sub graphs (except one node) of basic story graph U. Hence the structural difference between the graphs and the basic story graph U reduces the similarity score of those graphs to a large extent.

Hungarian Algorithm: In the result shown in the table above, it is observed that for Hungarian algorithm with α = 0.5 and α = 0.25, the similarity ranking of graphs is almost same. In all values of α, the story graphs E3, E9, E11, E12 and E13 are ranked lower in the similarity and E5, E6, E4, E8 and E7 graphs are ranked higher than other graphs. E10 graph is ranked higher than other graphs when α = 0.75 but otherwise ranked lower.

Also note that using this algorithm E2 is ranked much higher than E3 despite having equal number of dissimilar nodes and edges with U for all values of α. In case of E2, the dissimilar nodes are placed at the lowest level and in case of E3; the dissimilar nodes are placed at the top level of story graph. Hence we infer that in case of Hungarian algorithm, node location is more important than the number of dissimilar nodes. In other words, dissimilarity in the story structures in the higher level of the story graph is more dominant factor in deciding the similarity of the stories with the basic story graph. The lower ranking of E11 and E9, where there are additional nodes in the higher level of the story graph, also supports this inference.

Volgenant and Jonker Algorithm: From Table 1, we can see that as per Volgenant and Jonker Algorithm, the ranking of graphs E12 and E13 varies widely with different values of α, which gives little credibility of using this algorithm using the above parameters. In all values of α, E10 ranks much higher than other graphs and E1, E2, E3 and E9 stories are ranked lower which gives an indication that structural commonality is more important when this algorithm is applied. When α = 0.25, the two most dissimilar graphs with U (comparing number of common nodes, common edges and story structure), E12 and E13, are placed at the top. This result clearly indicates the invalidity of the results for α = 0.25.

Experiment 2:

The next set of experiments is conducted to check the effects of node attributes when the cost of individual node and edge operations are same. As stated before, a type attribute is defined for nodes, which indicates whether a node is a goal node, or an action node. Goal nodes have type attribute value = 1 and action nodes have type attribute = 2. In the similarity measure calculation, we put the cost function for node attribute as squared distance and set the weighing parameter for node attribute as 1. We do not consider any attribute weightage for edges since in this study we consider all edges are same and have the same weightage. The experimental result with the above parameter is shown below:

Algorithm Used	Cost of Node Insertion/ Deletion	Cost of edge Insertion/Deletion	Number of Node attributes used	Number of edge attributes used	Weighting factor α = 0.5	Weighting factor α = 0.75	Weighting factor α = 0.25
Beam Search	1	1	1	0	E5, E6, E10 E4, E14 E8, E11 E1, E9 E7, E13 E12 E3 E2	E5, E6, E10 E4, E11, E14 E8 E9 E1 E7 E3 E2 E12 E13	E5, E6, E10 E4, E14 E11 E8 E1, E7, E9 E3, E13 E2 E12
Hungarian	1	1	1	0	E5, E6 E4 E8 E7 E1 E14 E11 E10 E2 E9 E12, E13 E3	E5, E6 E4 E8 E10, E14 E11 E7 E1 E2 E9 E13 E3 E12	E5, E6 E4 E8 E7 E1 E14 E11 E2 E10 E13 E9 E12 E3
VJ algorithm	1	1	1	0	E10 E4 E5, E6 E7, E13, E14 E11 E8 E1, E9 E12 E3 E2	E10 E4 E5, E6 E14 E11 E8 E7 E1, E9 E3 E13 E2 E12	E10 E4 E13 E12 E11 E5, E6 E14 E7 E1, E9 E8 E2

Table 4 : Result with one node attribute and equal cost for individual node and edge operation

Beam Search Algorithm: In the result shown in Table 2 above, it is observed that for all three values of α, the ranking of graphs are more or less similar using Beam Search algorithm under the above parameter setting. E5, E6, E10, E4 and E14 stories rank higher whereas E3, E2, E13 and E12 are ranked lower. From the high ranking of E5, E6, E10 and E4, it is clear that the having the same story structure as U influenced the ranking of the stories mostly. Also having an additional sub-graph caused lower ranking of graph E3 and E2. Similarly lack of having the sub-graphs caused the lower ranking of E12 and E13. The lower ranking of graphs having the maximum structural difference with basic story graph U also supports our assumption that the structural commonality is the most influencing factor in determining the story ranking under the above parameter setting.

Hungarian Algorithm: From the result shown in Table 2 above, it is again noted that for α = 0.5 and α = 0.25, the similarity ranking of graphs is almost same using Hungarian algorithm. When α = 0.75, the top ranked order and the bottom ranked order of the graphs also remain almost same as for other two values of α, but the ranking of E10, E11 and E14 is much higher compared to the ranking of α = 0.5 and α = 0.25.

From the ranking obtained in the above setting, we do not find any specific influencing factor in determining similarity ranking of the story graphs. The higher ranking of E5 and E6 story graphs indicate structural commonality as the influencing factor but that assumption does not explain the lower ranking of E10 story which have the same structure as the base graph. Similarly if we consider the low number of different nodes influence the higher ranking of E4 and E8, which does not explain the lower ranking of E11. Hence from iMuse story comparison context, using Hungarian algorithm under the above parameter setting is not very helpful.

Volgenant and Jonker Algorithm: Like the previous experiment, we can see from Table 2 that story graphs E13 and E12 are ranked higher than other graphs when α = 0.25 which is inconsistent with the human reader ranking. Also E13 gets higher ranking when α = 0.5. Hence we disregard the ranking obtained from Volgenant and Jonker Algorithm under the above parameter setting when α <= 0.5

For α = 0.75, the ranking of the story graphs are somewhat more rational, but still does not indicate any dominant factor in ranking the story graphs.

Experiment 3:

The next set of experiments is conducted to check the effect of lower cost of individual node operation (compared to cost of individual edge operation) on similarity ranking when one node attribute is considered for similarity ranking (as in experiment 2). No edge attribute is considered. The experimental result with the above parameter is shown below:

Algorithm Used	Cost of Node Insertion/ Deletion	Cost of edge Insertion/Deletion	Number of Node attributes used	Number of edge attributes used	Weighting factor α = 0.5	Weighting factor α = 0.75	Weighting factor α = 0.25
Beam Search	0.75	1	1	0	E5, E6, E10 E4, E14 E8 E9 E1 E12, E13 E7 E11 E3 E2	E5, E6, E10 E4, E11, E14 E8 E9 E1 E7 E3 E2 E12 E13	E5, E6, E10 E4, E14 E11 E12, E13 E8 E1, E7, E9 E3 E2
Hungarian	0.75	1	1	0	E5, E6 E4 E8 E7 E1 E14 E11 E2 E10 E9 E12, E13 E3	E5, E6 E4 E8 E14 E7 E11 E1, E10 E2 E9 E13 E3, E12	E5, E6 E4 E8 E7 E1 E14 E11 E2 E10 E13 E12 E9 E3
VJ algorithm	0.75	1	1	0	E10 E4 E5, E6 E13 E7 E14 E11 E8, E12 E1, E9 E3 E2	E10 E4 E5, E6 E14 E11 E8 E7 E1, E9 E13 E3 E2 E12	E10 E4 E13 E12 E11 E5, E6 E14 E7 E3 E1, E9 E8 E2

Table 5 : Result with lower cost of individual node operation with one node attribute

Beam Search Algorithm: In the result presented in Table 3 above, it is observed that for all three values of α, E5, E6, E10, E4 and E14 are ranked higher when Beam Search algorithm is applied with the above parameter setting. But there are a lot of dissimilarities in the lower ranks of stories for different values of α. Notably, story graph E11, which has only one extra node at the top of the basic graph, is ranked much lower when α = 0.5.

Story graph E12 and E13 are also ranked differently for different values of α. The higher ranking of E12 and E13 story graphs also does not give much credibility to the ranking when α = 0.25 and α = 0.5. From the human reader perspective, E12 and E13 are among the most dissimilar stories with the basic graph U, hence we disregard the ranking obtained with Beam Search algorithm in the above parameter setting when α = 0.25 and α = 0.5.

The results obtained with α = 0.75 is more aligned with the human expectation – specially the higher ranking of E5, E6, E10, E4 and E11 indicates the high structural similarity with the basic graph is the most influencing factor in the ranking. Similarly the structural dissimilarity also ensures lower ranking of E2, E3, E12 and E13.

Hungarian Algorithm: In the result observed in Table 3 above, it is observed that the ranking of graphs for α = 0.5 and α = 0.25 are almost same when Hungarian algorithm is applied. Also there is not much difference in ranking when α = 0.75. For all three values of α, the graphs E5, E6, E4 and E8 are ranked higher and E9, E13, E3 and E12 are ranked lower. E10, which has all the nodes common with the base graph but the positions of two nodes interchanged, is ranked lower by the algorithm.

Higher ranking of E5, E6, E4 and E8 is a clear indication that having less number of uncommon nodes (all of them have only 1 different node) is an influencing factor, but from relative ranking of story graphs at the bottom also indicates that the positions of the nodes in the story graph that are different from the basic graph are very important. For example, E2 and E3 have same number of uncommon nodes with basic graph U, but E2 is ranked higher than E3; since the position of uncommon nodes in E2 are in lower level of the story graph compared to the position of the uncommon nodes in E3 which are in the top level. Also E9 has two uncommon additional nodes in the higher level compared to the base graph U which is the reason of its lower ranking than E2. Since story graph E3 has three uncommon nodes in the higher level, it is ranked lower than E9.

The ranking of E10 is also heavily penalized as the positions of the nodes that are interchanged are in the higher level of the story graph. The nodes whose positions are interchanged are connected with a large number of nodes; so changing the positions of the nodes created a large number of dissimilar edges with the base graph. Hence we can infer that structural dissimilarity in the higher level of the story graph along with number of dissimilar edges are the dominating factors in the story similarity ranking under the above parameter setting with Hungarian algorithm.

Volgenant and Jonker Algorithm: With the above parameter setting, the story graph E12 and E13 are placed higher in similarity ranking with the basic story graph U when α = 0.25, which is contrary to the human reader ranking. Hence we disregard the ranking of the story graphs provided by Volgenant and Jonker Algorithm when α = 0.25. Also higher ranking of E13 when α = 0.5 is also not aligned with human reader ranking. When α = 0.75, the ranking of the stories produced by Volgenant and Jonker Algorithm is most aligned with human reader ranking which is influenced by the low difference in the number of uncommon nodes between the story graph and the basic graph U.

Experiment 4:

In the next set of experiments, we interchange the cost of individual node and edge operation and rest of the parameters remain same as in experiment 3. That is the cost of individual node operation is now 1 whereas the cost of individual edge operation is set 0.75. The results are shown below:

Algorithm Used	Cost of Node Insertion/Deletion	Cost of edge Insertion/Deletion	Number of Node attributes used	Number of edge attributes used	Weighting factor α = 0.5	Weighting factor α = 0.75	Weighting factor α = 0.25
Beam Search	1	0.75	1	0	E11,E12, E14 E2, E4, E9 E7 E8 E1 E3 E13 E10 E5 E6	E5, E6, E10 E4, E11, E14 E8 E9 E1 E7 E3 E2 E13 E12	E5, E6, E10 E4, E14 E11 E8 E1, E7, E9 E3 E13 E2 E12
Hungarian	1	0.75	1	0	E5, E6 E4 E8 E7 E1 E14 E11 E10 E2 E9 E12 E13 E3	E5, E6 E4 E8 E10 E14 E11 E7 E1 E2, E9 E12 E3 E13	E5, E6 E4 E8 E7 E1 E14 E11 E2 E10 E9 E13 E12 E3
VJ algorithm	1	0.75	1	0	E10 E4 E5, E6 E14 E7 E11 E8 E13 E1, E9 E13 E3 E2	E10 E4, E5, E6 E14 E11 E8 E7 E1, E9 E3 E2 E12, E13	E10 E4 E13 E11 E5, E6 E12 E14 E7 E3 E1, E9 E8 E2

Table 6 : Result with lower individual edge edit operation with single node attribute

Beam Search Algorithm: In the result presented in Table 4 above, it is observed that for α = 0.5, E5 and E6 story graphs are ranked at the lowest level when Beam Search algorithm is applied which is inconsistent with the human reader ranking. Hence the set of results corresponding to α = 0.5 can be assumed to be not suitable for computing story similarities of iMuse stories.

For α = 0.75 and α = 0.25, the rank of the stories are almost similar. Hence in this case, the position of the nodes are not a deciding factor in calculating story similarities – the most influencing factor in this parameter setting is the structural commonality and less number of uncommon nodes.

Hungarian Algorithm: From the result in Table 4 above, it is again noted that for α = 0.5 and α = 0.25, the similarity ranking of graphs is almost same when Hungarian algorithm is applied. It is also noticed that with α = 0.5 and α = 0.25, the story graphs like E10 and E11 which have 0 or 1 dissimilar nodes with U are placed lower in the similarity ranking. For E10, the node positions are interchanged in the higher level whereas for E11, an additional node is added at the top most level. Compared to that story graph E1 and E7 have more additional nodes at the lower level with respect to U. Hence it indicates the changes in the higher level has higher penalizing effect on the similarity ranking of the graphs when α = 0.5 and α = 0.25. On the other hand α = 0.75, structural commonality is a more influencing factor in ranking the story graphs.

Volgenant and Jonker Algorithm: In the above setting, the ranking provided by Volgenant and Jonker Algorithm when α = 0.25, E13 is placed higher than most of the other story graphs. So we ignore the results obtained from this algorithm when α = 0.25. The ranking obtained when α = 0.75 and α = 0.5 are more or less similar in the higher order. The higher order ranking is indicative that the having the same structure as the basic graph (i.e. node replacements have lesser impact than node addition or deletion from the basic graph) and having less number of uncommon nodes with the basic graph are the most influencing factors in ranking the stories using this algorithm when α >= 0.5. The lower order of the ranking although does not indicate any specific influencing factor.

Experiment 5:

In this set of experiments, the node attributes are removed but the cost of individual node operation is made higher than the cost of individual edge operation i.e. the cost of individual node operation is set to 1 and cost of individual edge operation is set to 0.75. No edge attribute is considered. The experimental result with the above parameter is shown below:

Algorithm Used	Cost of Node Insertion/Deletion	Cost of edge Insertion/Deletion	Number of Node attributes used	Number of edge attributes used	Weighting factor α = 0.5	Weighting factor α = 0.75	Weighting factor α = 0.25
Beam Search	1	0.75	0	0	E5, E6 E10 E4, E11, E14 E8 E1, E7, E9 E3 E2 E12, E13	E5, E6 E10 E4, E11, E14 E8 E1, E7, E9 E3 E2 E12, E13	E5, E6 E4, E11, E14 E8 E1, E7, E9, E10 E12, E13 E3 E2
Hungarian	1	0.75	0	0	E5, E6 E4, E8 E7 E1 E2 E10 E14 E11 E9 E3 E12, E13	E5, E6 E4, E8 E10 E7, E14 E1 E2 E11 E9 E3 E12, E13	E5, E6 E4, E8 E7 E1 E2 E14 E10 E12, E13 E9 E3 E11
VJ algorithm	1	0.75	0	0	E10 E5, E6 E4 E8 E11 E14 E7 E1 E9 E12, E13 E3 E2	E10 E5, E6 E4 E8 E11 E14 E7 E1 E9 E3 E2 E12, E13	E10 E12, E13 E4 E5, E6 E8 E7 E11 E1 E14 E9 E3 E2

Table 7 : Result with lower individual node edit operations and no node attributes

Beam Search Algorithm: In the result shown in Table 5 above, it is observed that in this parameter setting, for α = 0.5 and α = 0.75, the ranking of the graphs are almost same when Beam Search algorithm is applied. Even with α = 0.25, there is not much difference in ranking of graphs except story graph E10 is ranked lower than story graph E8. In the story graph E8, one node (which was there in U) is removed and a different node (which is not there in U) is placed in the same position in the story graph where as in story graph E10, two common nodes have exchanged their places. However, the result suggests that with no node attributes, and higher weightage of individual node operations, the effect of α is much lower in ranking of the story graphs. The most influencing factors in ranking the stories are having the same structure as the basic graph (i.e. node replacements have less impacts than node addition or deletion) and having less number of uncommon nodes with the basic graph. The node positions are not very dominating factor in ranking the stories.

Hungarian Algorithm: In this case, the ranking of the story graphs for α = 0.5 and α = 0.25 matches for the top ranked graphs. On the other hand, the ranking of the story graphs matches in the lower order when α = 0.5 and α = 0.75. From Table 5, it is observed that in the above parameter setting, the E3, E9 and E11 story graphs are ranked lower in all values of α i.e. presence of an additional node at the top severely affects the similarity ranking of the graphs in this setting when Hungarian algorithm is applied. This assumption is further supported by relative higher ranking of E1 and E2 story graphs, which have more number of additional story nodes in the bottom level in comparison with the basic story graph. Clearly the absence of the story nodes make less impact in ranking when α = 0.25.

Volgenant and Jonker Algorithm: From the result in Table 5, one can find that, in this parameter setting, the ranking of the story graphs are almost similar when α = 0.5 and α = 0.75 when Volgenant and Jonker algorithm is applied. The number of uncommon nodes seems to be the dominating factor in story ranking under this parameter setting when α = 0.5 and α = 0.75. Also the lower ranking of E9 compared to E1 and E7 can indicate that having more number of additional node does not impact as much as having more number of additional node in the high level.

The ranking for α = 0.25 is disregarded since it places the story graphs E12 and E13 among the high similarity graphs with basic graph U.

Experiment 6:

In this set of experiments, all the parameters remain same as experiment 5 but the cost of individual node operation is made lower than the cost of individual edge operation i.e. the cost of individual node operation is set to 0.75 and cost of individual edge operation is set to 1. No node or edge attribute is considered. The experimental result with the above parameter is shown below:

Algorithm Used	Cost of Node Insertion/ Deletion	Cost of edge Insertion/Deletion	Number of Node attributes used	Number of edge attributes used	Weighting factor α = 0.5	Weighting factor α = 0.75	Weighting factor α = 0.25
Beam Search	0.75	1	0	0	E5, E6 E4, E11, E14 E8 E1, E7, E9, E10 E3 E2 E12, E13	E5, E6 E10 E4, E11, E14 E8 E1, E7, E9 E3 E2 E12, E13	E5, E6 E4, E11, E14 E8, E12, E13 E1, E7, E9, E10 E3 E2
Hungarian	0.75	1	0	0	E5, E6 E4, E8 E7 E1 E2 E14 E10 E12, E13 E9 E11 E3	E5, E6 E4, E8 E7 E1, E10 E14 E2 E11 E9 E3 E12, E13	E5, E6 E4, E8 E7 E1 E2 E14 E10 E12, E13 E9 E3 E11
VJ algorithm	0.75	1	0	0	E10 E4 E5, E6 E8, E12, E13 E11 E7 E14 E1 E9 E3 E2	E10 E5, E6 E4 E8 E11 E14 E7 E1 E9 E3 E12, E13 E2	E12, E13 E10 E4 E5, E6 E8 E7 E11 E1 E14 E9 E3 E2

Table 8 : Result of story comparison with lower cost of individual node operations with no attributes

Beam Search Algorithm: From Table 6, it is observed that in this parameter setting, for α = 0.5 and α = 0.75, the ranking of the graphs are almost same except E10 which indicates cost of interchanging the nodes is more penalizing when α = 0.5. Otherwise for α >= 0.5, having the same graph structure and less number of uncommon nodes are the most influencing factor.

For α = 0.5 and α = 0.25, the ranking of the graphs are almost same but except story graph E12 and E13. We disregard ranking obtained from α = 0.25 since the higher ranking of E12 and E13 are contrary to the human reader ranking.

Hungarian Algorithm: From the observed result in Table 6, we can find that the ranking of the stories are same when α = 0.5 and α = 0.25 whereas for α = 0.75, the ranking of the story graphs in the lower order is considerably different especially for E10 and E11. Since the ranking of the stories E12 and E13 are placed higher than story graph E11 when α = 0.5 and α = 0.25, we disregard the ranking of the stories in those two cases.

E10 and E11 are placed at much higher position in the ranking when α = 0.75. However, since E11, E9 and E3 are placed lower than E1 and E2, it indicates having more structural differences in the higher level of the story graph have more impact than having structural differences in the lower level of the story.

Volgenant and Jonker Algorithm: We disregard the result set using this algorithm when α <= 0.5, the story graphs E12 and E13 placed at a higher ranks. When α = 0.75, having less number of uncommon nodes with the basic graph is the most dominating factor.

5.1 Observation

From the above set of results, it is clear that different factors determine the ranking of the story graphs under different parameter setting for different algorithms and sometimes it is difficult to determine the most influencing factor. In many cases, for two different values of α, almost identical ranking of story graphs are obtained which is significantly different for the result obtained from the other value of α. Some of the results could be directly rejected since they are in direct conflict with the human reader ranking.

In the following section, we collated the findings obtained from the same algorithm under different parameter settings. We have identified the similar rankings in the same color in the below tables – each color represents a ranking (the name of the story identifiers are organized as per their similarity ranking with the basic graph U). The cells, which are marked in red, represent discarded results – since in those cases, the story graphs E12 and E13 are placed much higher than other graphs. The cells that are not colored represent ranking of the stories, which are not matched by any other results, obtained by the algorithm under different parameter settings.

In the below table, we present the experimental result for ***Beam Search algorithms*** for different parameters.

Cost of Node Insertion/ Deletion	**Cost of edge Insertion/ Deletion**	**No. of Node attributes used**	**Weighting factor α = 0.5**	**Weighting factor = 0.75**	**Weighting factor α = 0.25**
1	1	0	E5, E6 E10 E11, E14, E4 E8 E1, E7, E9 E3 E12, E13,E2	E5, E6 E10 E4, E11, E14 E8 E1, E7, E9 E3 E2 E12, E13	E5, E6 E4, E11, E14 E8 E1, E7, E9, E10, E12, E13 E3 E2

Cost of Node Insertion/ Deletion	**Cost of edge Insertion/ Deletion**	**No. of Node attributes used**	**Weighting factor α = 0.5**	**Weighting factor = 0.75**	**Weighting factor α = 0.25**
1	1	1	E5, E6, E10 E4, E14 E8, E11 E1, E9 E7, E13 E12 E3 E2	E5, E6, E10 E4, E11, E14 E8 E9 E1 E7 E3 E2 E12 E13	E5, E6, E10 E4, E14 E11 E8 E1, E7, E9 E3, E13 E2 E12
1	0.75	1	E11,E12, E14 E2, E4, E9 E7 E8 E1 E3 E13 E10 E5 E6	E5, E6, E10 E4, E11, E14 E8 E9 E1 E7 E3 E2 E13 E12	E5, E6, E10 E4, E14 E11 E8 E1, E7, E9 E3 E13 E2 E12
0.75	1	1	E5, E6, E10 E4, E14 E8 E9 E1 E12, E13 E7 E11 E3 E2	E5, E6, E10 E4, E11, E14 E8 E9 E1 E7 E3 E2 E12 E13	E5, E6, E10 E4, E14 E11 E12, E13 E8 E1, E7, E9 E3 E2
1	0.75	0	E5, E6 E10 E4, E11, E14 E8 E1, E7, E9 E3 E2 E12, E13	E5, E6 E10 E4, E11, E14 E8 E1, E7, E9 E3 E2 E12, E13	E5, E6 E4, E11, E14 E8 E1, E7, E9, E10 E12, E13 E3 E2

Cost of Node Insertion/ Deletion	**Cost of edge Insertion/ Deletion**	**No. of Node attributes used**	**Weighting factor α = 0.5**	**Weighting factor = 0.75**	**Weighting factor α = 0.25**
0.75	1	0	E5, E6 E4, E11, E14 E8 E1, E7, E9, E10 E3 E2 E12, E13	E5, E6 E10 E4, E11, E14 E8 E1, E7, E9 E3 E2 E12, E13	E5, E6 E4, E11, E14 E8, E12, E13 E1, E7, E9, E10 E3 E2

Table 9 : Beam search algorithm results with different parameters

Legends:

(red)	Discarded Ranking
(light green)	Ranking Order 1
(yellow)	Ranking Order 2 (Similar to Ranking Order 1)
(violet)	Ranking Order 3
(light blue)	Ranking Order 4
(white)	Ranking Order 5 (not similar with any other ranking order in the table)

From the table above, we can find the cells marked in light green and yellow provide the most consistent result with the human reader rankings. The ranking of the stories in yellow and green cells closely resemble each other. In both cases, E5, E6, E10, E4, E11 and E14 story graphs are ranked in the top i.e. having the highest similarity with the basic story graph. Story graphs E12, E13, E2 and E3 are in the lower rank i.e. they are most dissimilar with the basic story graphs. The ranking obtained in the light blue and violet cells are less preferred since they rank E3 and E2 story graphs after E13 and E12.

Overall, we can say that when weighting factor α = 0.75, Beam search algorithm produces the most consistent result with human reader ranking. Also we can note that when no node and edge attributes are used, α = 0.5 also produces the same ranking with the sample story graphs.

Let us now collate the results obtained from ***Hungarian algorithm*** for different parameters.

Cost of Node Insertion /Deletion	**Cost of edge Insertion/ Deletion**	**Number of Node attributes used**	**Weighting factor α = 0.5**	**Weighting factor α = 0.75**	**Weighting factor α = 0.25**
1	1	0	E5, E6 E4, E8 E7 E1 E2 E10, E14 E9, E11 E12, E13 E3	E5, E6 E4, E8 E10 E7 E1 E14 E2 E11 E9 E3 E12, E13	E5, E6 E4, E8 E7 E1 E2 E14 E10 E12,E13 E9 E3 E11
1	1	1	E5, E6 E4 E8 E7 E1 E14 E11 E10 E2 E9 E12, E13 E3	E5, E6 E4 E8 E10, E14 E11 E7 E1 E2 E9 E13 E3 E12	E5, E6 E4 E8 E7 E1 E14 E11 E2 E10 E13 E9 E12 E3
1	0.75	1	E5, E6 E4 E8 E7 E1 E14 E11 E10 E2 E9 E12 E13 E3	E5, E6 E4 E8 E10 E14 E11 E7 E1 E2, E9 E12 E3 E13	E5, E6 E4 E8 E7 E1 E14 E11 E2 E10 E9 E13 E12 E3

Cost of Node Insertion /Deletion	**Cost of edge Insertion/ Deletion**	**Number of Node attributes used**	**Weighting factor α = 0.5**	**Weighting factor α = 0.75**	**Weighting factor α = 0.25**
0.75	1	1	E5, E6 E4 E8 E7 E1 E14 E11 E2 E10 E9 E12, E13 E3	E5, E6 E4 E8 E14 E7 E11 E1, E10 E2 E9 E13 E3, E12	E5, E6 E4 E8 E7 E1 E14 E11 E2 E10 E13 E12 E9 E3
0.75	1	0	E5, E6 E4, E8 E7 E1 E2 E14 E10 E12, E13 E9 E11 E3	E5, E6 E4, E8 E7 E1, E10 E14 E2 E11 E9 E3 E12, E13	E5, E6 E4, E8 E7 E1 E2 E14 E10 E12, E13 E9 E3 E11
1	0.75	0	E5, E6 E4, E8 E7 E1 E2 E10 E14 E11 E9 E3 E12, E13	E5, E6 E4, E8 E10 E7, E14 E1 E2 E11 E9 E3 E12, E13	E5, E6 E4, E8 E7 E1 E2 E14 E10 E12, E13 E9 E3 E11

Table 10 : Hungarian algorithm results with different parameters

Legends:

	Discarded Ranking
	Ranking Order 1
	Ranking Order 2
	Ranking Order 3
	Ranking Order 4
	Ranking Order 5
	Ranking Order 6 (not similar with any other ranking order in the table)

From the table above, we can find the cells marked in light blue provide the most consistent result with the human reader rankings since they put story graphs E12 and E13 at the bottom of their ranking all the times. It is also observed that the most consistent result with Hungarian algorithm is observed when weighting factor $\alpha = 0.75$ and number of node attributes = 0 and cost of node individual node insertion is greater than or equal to the cost of edge insertion/deletion.

Using this algorithm, E5, E6, E4, and E8 story graphs are ranked in the top and story graphs E2, E11, E9, E3, E12 and E13 are in the lower order in the overall ranking. As already observed, in general the story graphs having more structural dissimilarity in the higher level are more penalized in the ranking.

Let us now collate the results obtained from ***Volgenant and Jonker algorithm*** for different parameters.

Cost of Node Insertion /Deletion	Cost of edge Insertion/ Deletion	Number of Node attributes used	Weighting factor α = 0.5	Weighting factor α = 0.75	Weighting factor α = 0.25
1	1	0	E10 E4, E5, E6 E8 E12, E13 E11 E7, E14 E1 E9 E3 E2	E10 E5, E6 E4 E8 E11 E14 E7 E1 E9 E3 E2 E12, E13	E12, E13 E10 E4 E5, E6 E8 E7 E11 E1 E14 E9 E3 E2
1	1	1	E10 E4 E5, E6 E7, E13, E14 E11 E8 E1, E9 E12 E3 E2	E10 E4 E5, E6 E14 E11 E8 E7 E1, E9 E3 E13 E2 E12	E10 E4 E13 E12 E11 E5, E6 E14 E7 E1, E9 E8 E2
1	0.75	1	E10 E4 E5, E6 E14 E7 E11 E8 E13 E1, E9 E13 E3 E2	E10 E4, E5, E6 E14 E11 E8 E7 E1, E9 E3 E2 E12, E13	E10 E4 E13 E11 E5, E6 E12 E14 E7 E3 E1, E9 E8 E2

Cost of Node Insertion /Deletion	Cost of edge Insertion/ Deletion	Number of Node attributes used	Weighting factor α = 0.5	Weighting factor α = 0.75	Weighting factor α = 0.25
0.75	1	1	E10 E4 E5, E6 E13 E7 E14 E11 E8, E12 E1, E9 E3 E2	E10 E4 E5, E6 E14 E11 E8 E7 E1, E9 E13 E3 E2 E12	E10 E4 E13 E12 E11 E5, E6 E14 E7 E3 E1, E9 E8 E2
1	0.75	0	E10 E5, E6 E4 E8 E11 E14 E7 E1 E9 E12, E13 E3 E2	E10 E5, E6 E4 E8 E11 E14 E7 E1 E9 E3 E2 E12, E13	E10 E12, E13 E4 E5, E6 E8 E7 E11 E1 E14 E9 E3 E2
0.75	1	0	E10 E4 E5, E6 E8, E12, E13 E11 E7 E14 E1 E9 E3 E2	E10 E5, E6 E4 E8 E11 E14 E7 E1 E9 E3 E12, E13 E2	E12, E13 E10 E4 E5, E6 E8 E7 E11 E1 E14 E9 E3 E2

Table 11: Volgenant and Jonker algorithm results with different parameters

Legends:

(red)	Discarded Ranking
(yellow)	Ranking Order 1
(pink)	Ranking Order 2

From the table above, we can find from the discarded results that Volgenant and Jonker Algorithm is only suitable for story comparison when weighting factor $\alpha = 0.75$. However, with $\alpha = 0.75$, two sets of ranking, represented by yellow and pink color respectively, have emerged. To match the human reader ranking, we will only consider those rankings produced by the algorithm as valid which ranks story graph E12 and E13 in the lowest ranking position.

5.2 Comparison With Human Reader Observation

From the above discussion, it is clear that the Beam Search, Hungarian and Volgenant and Jonker Algorithms cannot be used for story comparison purpose for all parameter settings for sample stories used in iMuse. However, these algorithms are able to produce some valid ranking results under certain parameter settings that are equivalent to the similarity ranking provided human readers.

Let us have a look at the valid rankings produced by the above three algorithms and match them with the inputs obtained from human readers in section 4.5:

Beam Search **Weighting factor = 0.75**	**Hungarian** **Weighting factor α = 0.75 and Number of Node attributes used = 0**	**Volgenant and Jonker Algorithm** **Weighting factor α = 0.75 and Number of Node attributes used = 0**
E5, E6 E10 E4, E11, E14 E8 E1, E7, E9 E3 E2 E12, E13	E5, E6 E4, E8 E10 E7 E1 E14 E2 E11 E9 E3 E12, E13	E10 E5, E6 E4 E8 E11 E14 E7 E1 E9 E3 E2 E12, E13

Table 12 : List of valid results produced by different algorithms

Legends:

(green)	Ranking Order from Beam Search Algorithm which is most aligned with human ranking
(light blue)	Ranking Order from Hungarian Algorithm which is most aligned with human ranking
(yellow)	Ranking Order from Volgenant and Jonker Algorithm which is most aligned with human ranking

We consider the results to be valid and aligned with the human reader ranking because:

1. In all the above results, story graphs E12 and E13 are placed at the bottom which is aligned with observation (a) in section 4.5
2. Story graphs E4, E5, E6, E10 and E11 are among the top most similar graphs in comparison with basic graph U (Exception: in Hungarian algorithm, E11 is placed at much lower position), which is aligned with human reader observation (b) in section 4.5. The exception of result in Hungarian algorithm can be attributed to the

fact that the algorithm gives more importance in maintaining same the root level objectives or main story plot points while comparing the stories rather than trivial differences in the detail story actions.

3. Story graph E7 has higher ranking than story graph E1 (Exception: E7 is placed at the same level with E1 for Beam search algorithm) which is aligned with human reader observation (c)
4. Story graph E9 is placed in higher ranking than story graph E3, which is aligned with human reader observation (d).
5. Story graph E11 is placed in higher ranking than story graph E9, which is aligned with human reader observation (e).

One notable anomaly between the human reader ranking and the ranking obtained from the algorithms is that story graph E1 is placed higher than story graph E2 by all the algorithms. If we compare the total number of nodes of the basic story graph U and story graph E1 and E2, we find that E1 has two extra nodes whereas E2 has three extra nodes than the basic story graph. Also we notice that in E2, the nodes are more mutually interconnected compared to E1, which implies that during the transformation, more number of edges got changed for E2. In E2, more details are added regarding insurance status of the vehicle whereas in E1, the story contains more details about more different factors like influence of alcohol, presence of illegal arms and registration status of vehicle. Hence from human reader perspective, E1 is more dissimilar with basic graph U than E2.

We attribute the anomaly in the result obtained from the algorithms to the fact that in our study we are not considering the semantic similarity between the nodes (E2 nodes are more semantically similar than E1 nodes). In our future work, we hope to have a more aligned result with the human ranking by including a semantic similarity factor while computing the similarity score between stories.

5.3 Conclusions And Future Work

The experimental results produced by using different graph edit distance algorithms can be particularly useful in iMuse story comparison context since they indicate the factors that are influencing the similarity ranking of the stories. The experiments were conducted as part of preliminary study to check the applicability of graph edit distance algorithms in the story comparison context. From the above experimental results, we can safely declare that with some parameter optimization, the Beam Search, Hungarian and Volgenant and Jonker algorithms can rank the stories in a reasonably logical manner based on their similarity with a base graph.

However, the above results also point to the scope of future work in this area. For the preliminary study, we used a very small story graph set, hence the indications and assumptions obtained from the results need to be substantiated by conducting the experiments with bigger set of data. For the preliminary study, we did not consider any edge attributes for similarity computation. The plot fragments or pre-conditions can be represented as the edge attribute for more complex story representation models. Hence in future, we would like to consider edge attributes with different weightages for story comparison. Besides, the algorithms provide the flexibility to decide whether we would like to rank stories based on more matching nodes or more matching edges between two stories. Hence this opens up a great potential for parameter optimization that are required for iMuse story model. Also as mentioned in the previous section, including a semantic similarity factor for similarity computation will be one of the key focuses in our future work.

Lastly, finding the similarities between stories can be dependent on the domain of the stories. For some domains, having the main outline or nodes may be more important irrespective of their order where as in other domains the operational details may be more important while deciding the similarity between stories. The parameter setting applied for one domain may not be used as baseline for other domain stories. We can also conduct similar experiments on stories from different domains in future and substantiate our findings.

It must be noted that finding story similarity has huge potential as no pioneering work is produced in this field so far. The challenge is more aggravated due to the lack of proper tools and lack of any generic approach in this field. Besides, more research is required about defining the required story attributes and the discriminative features of a story graph for a domain. New annotation techniques to enrich stories can help the researchers to achieve better experimental results. With more coordinated and consistent effort from researchers in various interdisciplinary fields, better results can be achieved.

REFERENCES

[1] R.J.S. Paul, M.D.J.Mcneil, D.K.Charles, D.M.G. McSherry, P.J.Morrow. "Real-time Planning for Interactive Storytelling." Proceedings of the 9th Irish Workshop on Computer Graphics. 2009.

[2] Graesser, Arthur C., Kathy L. Lang, and Richard M. Roberts. "Question answering in the context of stories." Journal of experimental psychology. General 120.3 (1991): 254-277.

[3] Riedl, Mark O., and Brian O'Neill. "Computer as audience: A strategy for artificial intelligence support of human creativity." Computational Creativity Support Workshop at CHI'09. 2009.

[4] Riedl, Mark O., Jonathan P. Rowe, and David K. Elson. "Toward intelligent support of authoring machinima media content: story and visualization." Proceedings of the 2nd international conference on INtelligent TEchnologies for interactive enterTAINment. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[5] Ram, Ashwin. "AQUA: Questions that drive the explanation process." (1993).

[6] Medler, Ben, and Brian Magerko. "Scribe: A tool for authoring event driven interactive drama." *Technologies for Interactive Digital Storytelling and Entertainment*. Springer Berlin Heidelberg, 2006. 139-150.

[7] Iurgel, Ido A. "Cyranus–an authoring tool for interactive edutainment applications." Technologies for E-Learning and Digital Entertainment. Springer Berlin Heidelberg, 2006. 577-580.

[8] Harel, David. "On visual formalisms." Communications of the ACM 31.5 (1988): 514-530.

[9] O'Neill, Brian, and Mark Riedl. "Supporting human creative story authoring with asynthetic audience." Proceedings of the seventh ACM conference on Creativity and cognition. ACM, 2009.

[10] Skorupski, James, Lakshmi Jayapalan, Sheena Marquez and Micheal Mateas. "Wide ruled: a friendly interface to author-goal based story generation." Virtual Storytelling. Using Virtual Reality Technologies for Storytelling. Springer Berlin Heidelberg, 2007. 26-37.

[11] Winbladh, Kristina, Hadar Ziv, and Debra J. Richardson. "iMuse: interactive model-based use-case and storytelling environment." Proceedings of the eighteenth ACM SIGSOFT international symposium on Foundations of software engineering. ACM, 2010.

[12] Richard C. Holt, Andy Schürr, Susan Elliott Sim, Andreas Winter. "GXL: A graph-based standard exchange format for reengineering." Science of Computer Programming 60.2 (2006): 149-170.

[13] http://www.fhnw.ch/wirtschaft/iwi/gmt

[14] K. Riesen, S. Emmenegger and H. Bunke. "A Novel Software Toolkit for Graph Edit Distance Computation. " In W.G. Kropatsch et al., editors, Proc. 9th Int. Workshop on Graph Based Representations in Pattern Recognition, LNCS 7877, 142–151, 2013

[15] Hart, P.E., Nilsson, N.J., Raphael, B.: "A formal basis for the heuristic determination of minimum cost paths." IEEE Transactions of Systems, Science, and Cybernetics 4(2), 100–107 (1968)

[16] Neuhaus, Michel, Kaspar Riesen, and Horst Bunke. "Fast suboptimal algorithms for the computation of graph edit distance." Structural, Syntactic, and Statistical Pattern Recognition. Springer Berlin Heidelberg, 2006. 163-172.

[17] Riesen, Kaspar, Michel Neuhaus, and Horst Bunke. "Bipartite graph matching for computing the edit distance of graphs." Graph-Based Representations in Pattern Recognition. Springer Berlin Heidelberg, 2007. 1-12.

[18] Riesen, Kaspar, and Horst Bunke. "Approximate graph edit distance computation by means of bipartite graph matching." Image and Vision Computing 27.7 (2009): 950-959.

[19] Fankhauser, Stefan, Kaspar Riesen, and Horst Bunke. "Speeding up graph edit distance computation through fast bipartite matching." Graph-based representations in pattern recognition. Springer Berlin Heidelberg, 2011. 102-111.

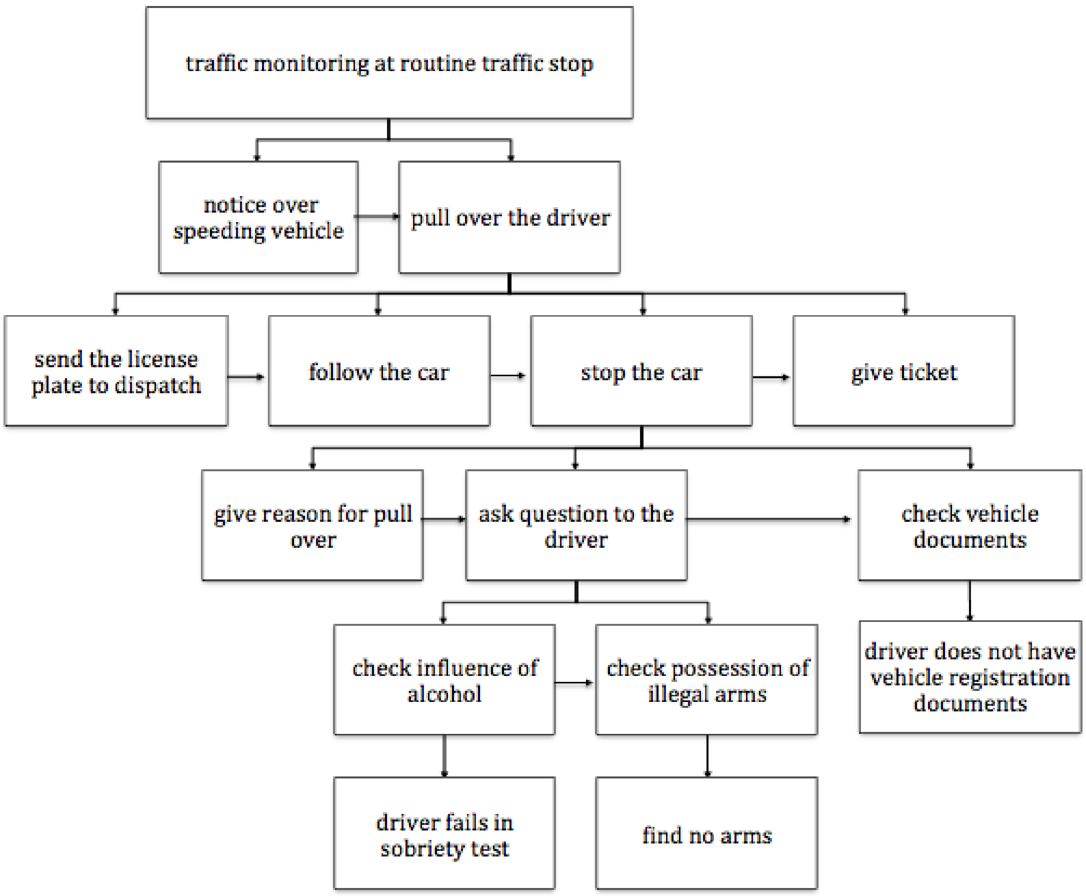
[20] Munkres, James. "Algorithms for the assignment and transportation problems." Journal of the Society for Industrial & Applied Mathematics 5.1 (1957): 32-38.

[21] Jonker, Roy, and Anton Volgenant. "A shortest augmenting path algorithm for dense and sparse linear assignment problems." Computing 38.4 (1987): 325-340.

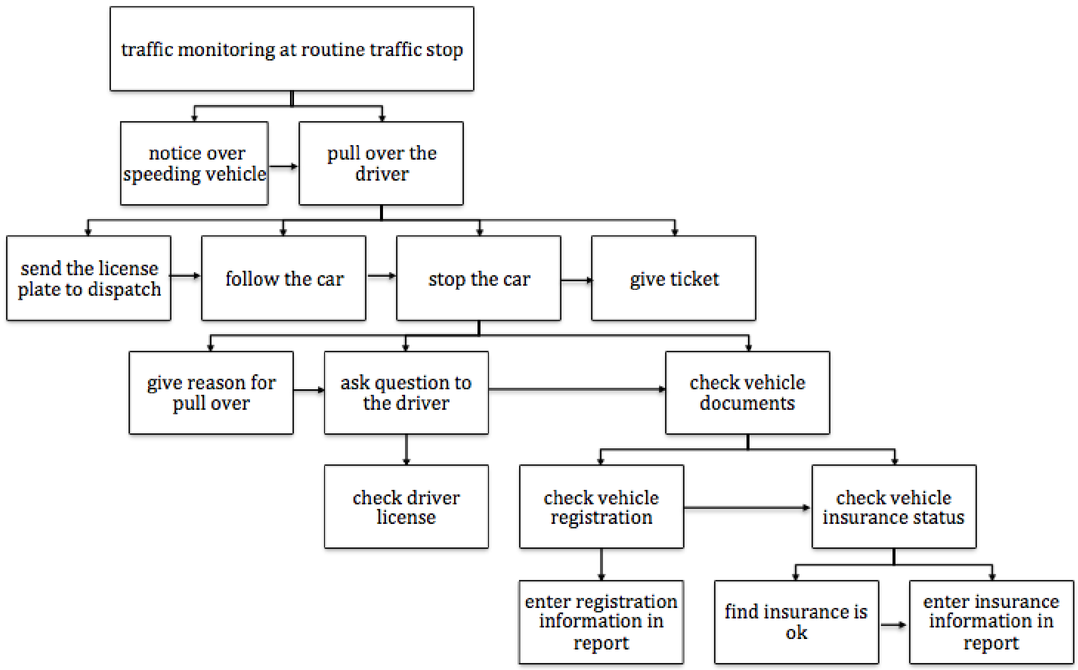
Appendix A

VISUAL REPRESENTATION OF SMALLER STORY GRAPHS

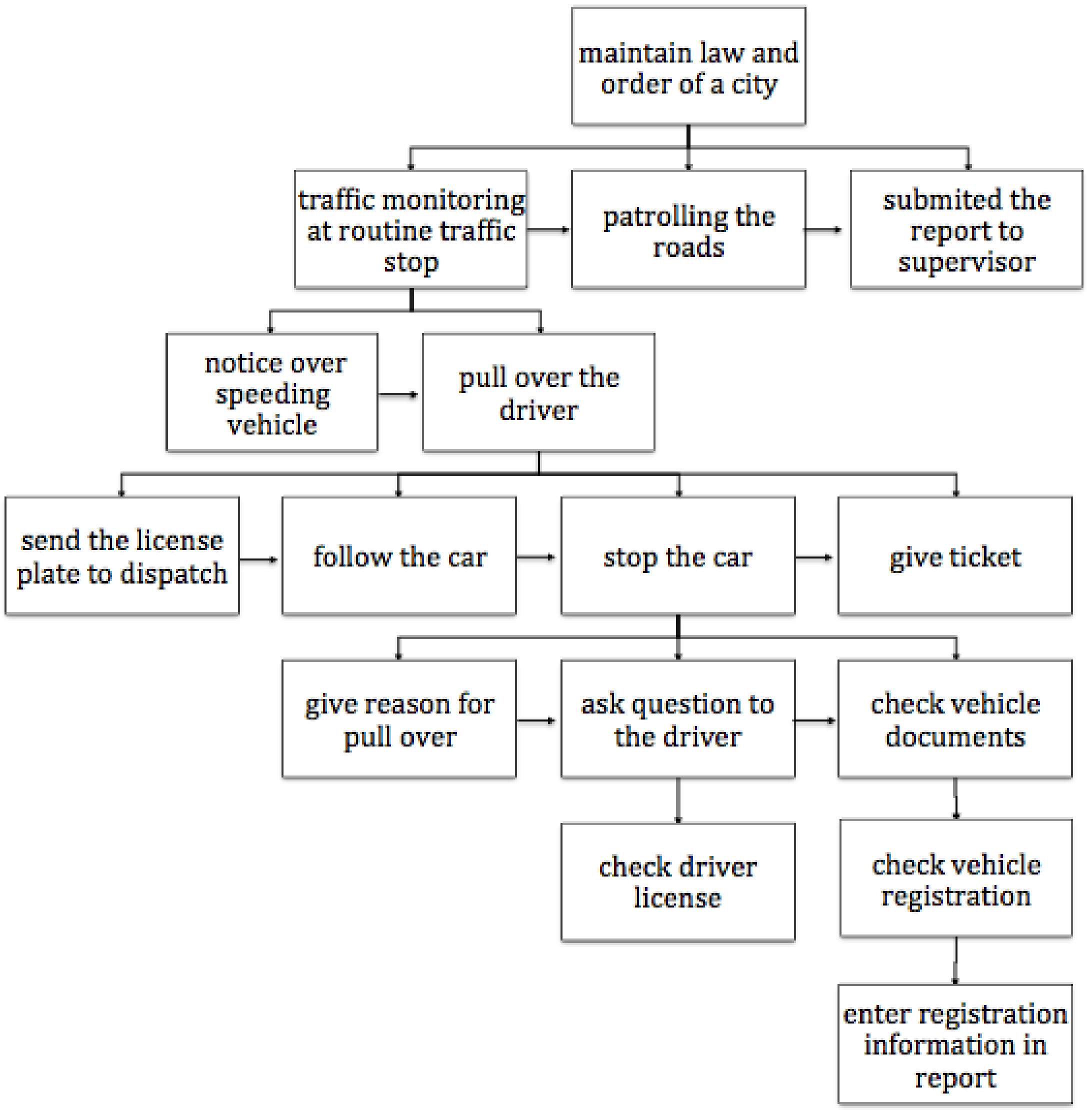
Smaller Story E1:



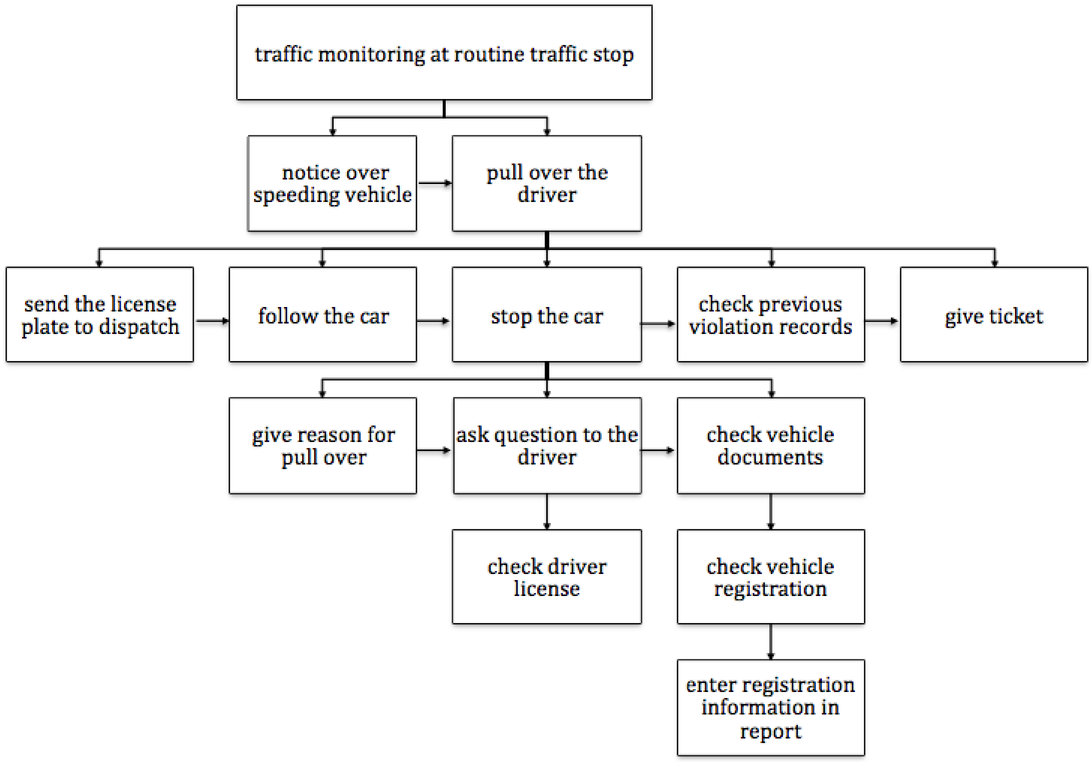
Smaller Story E2:



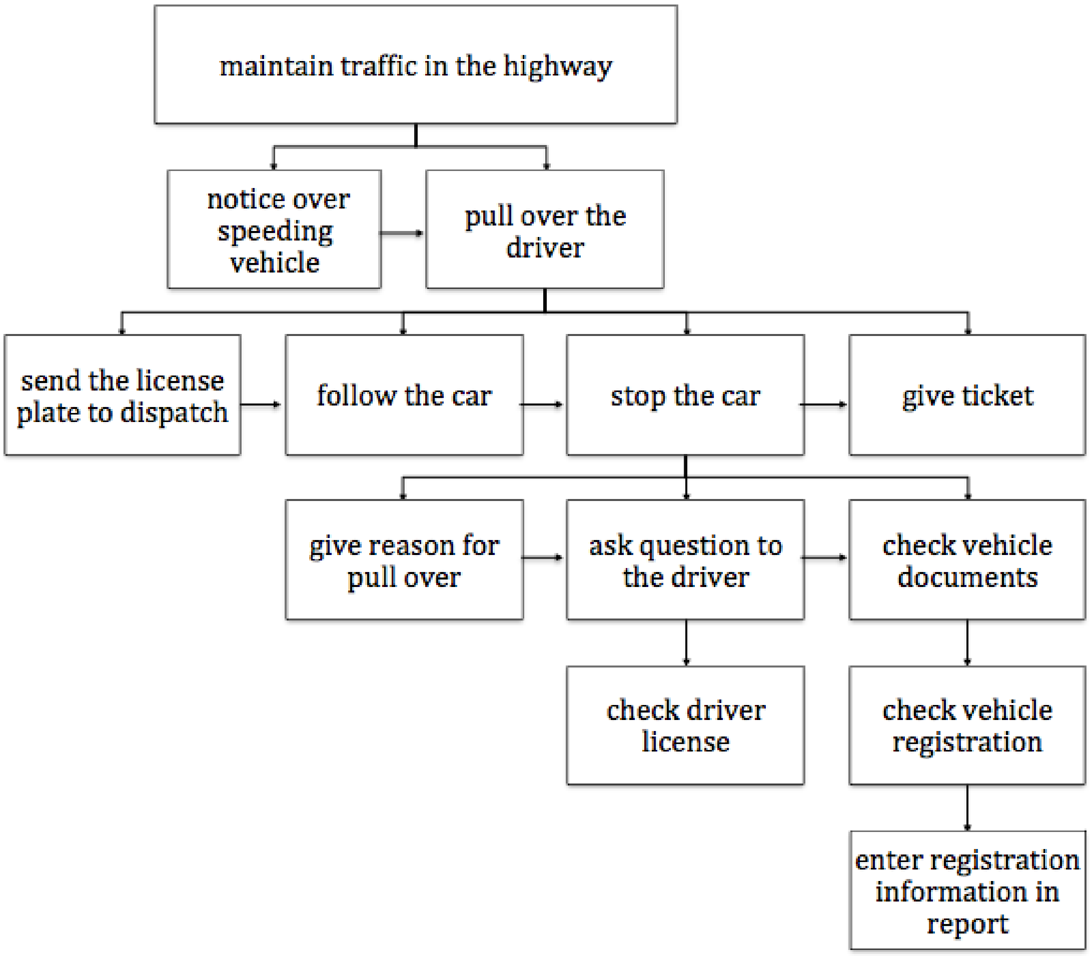
Smaller Story E3:



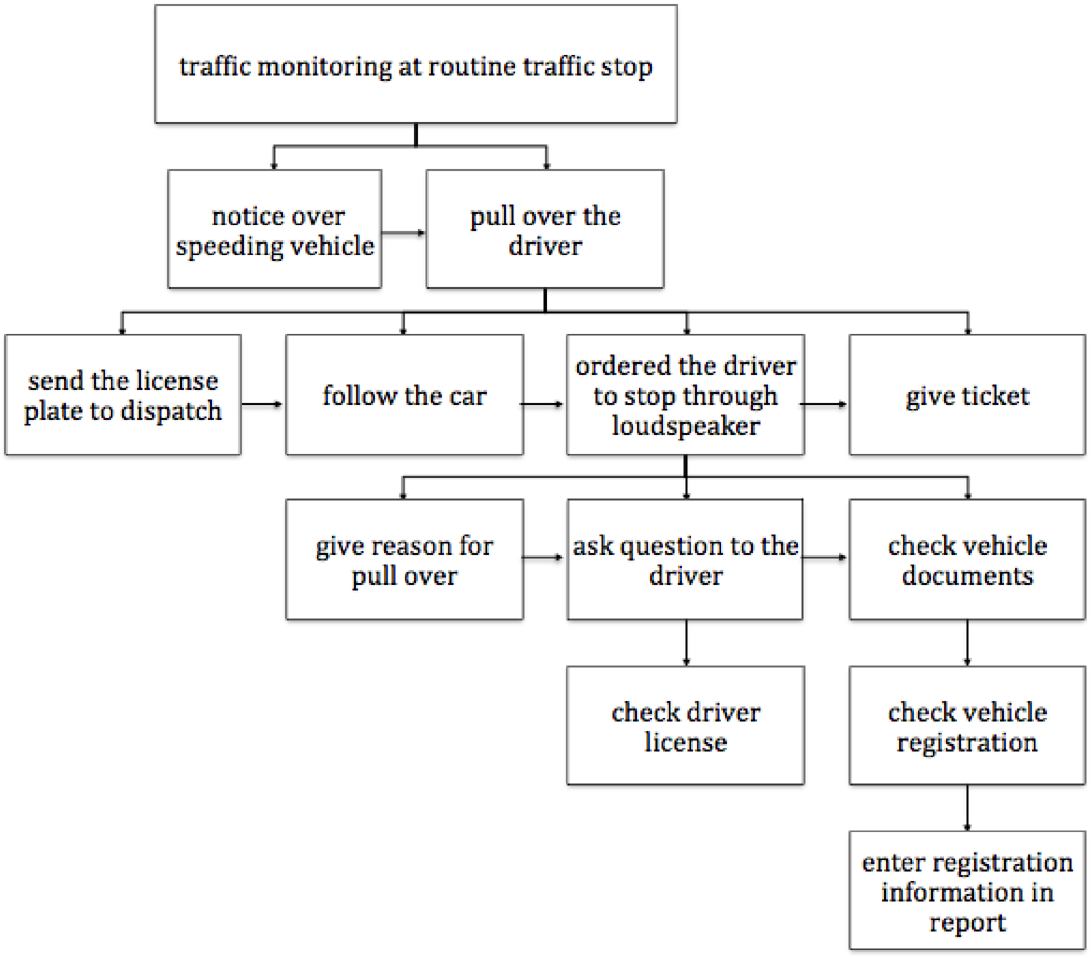
Smaller Story E4:



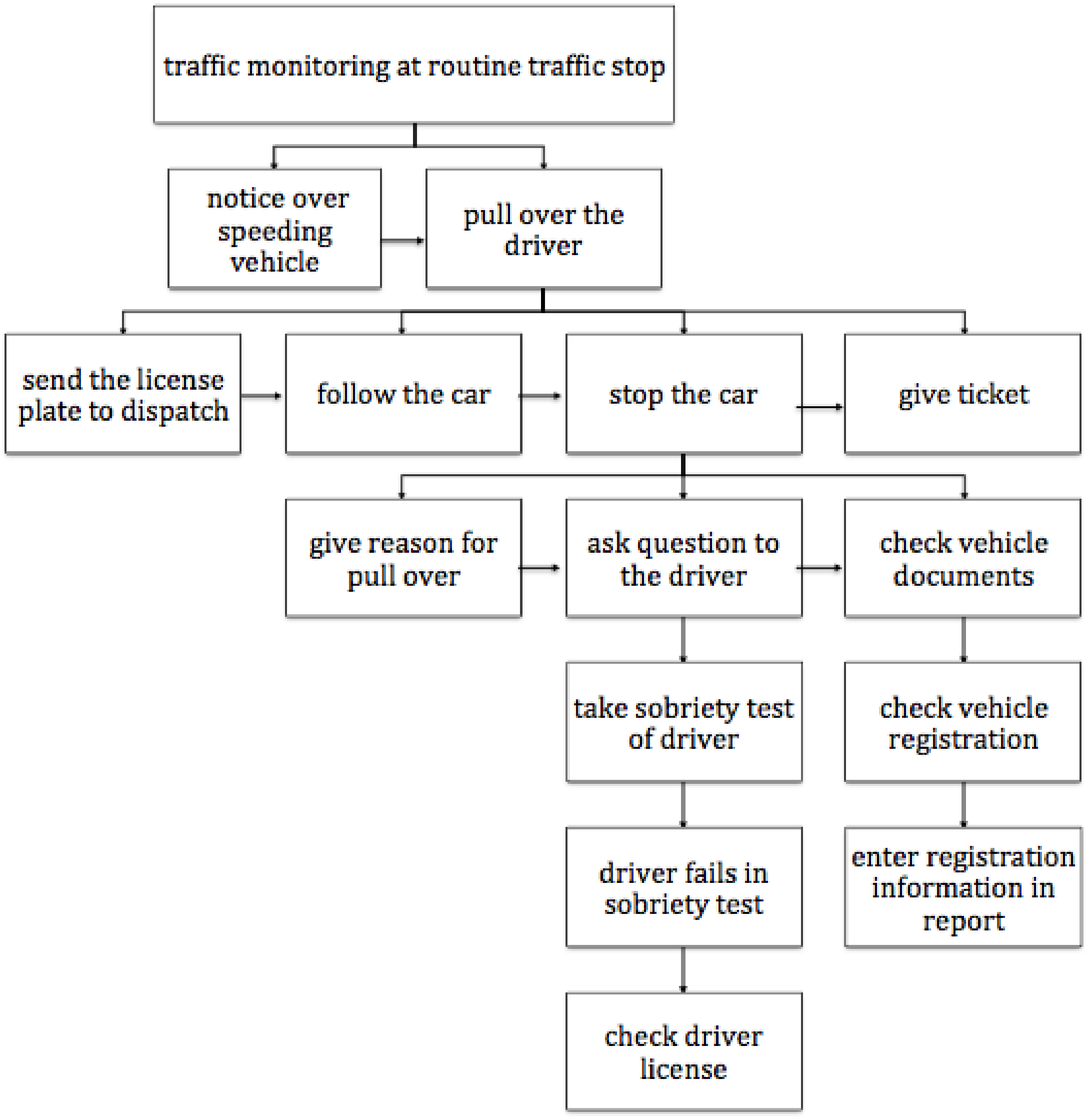
Smaller Story E5:



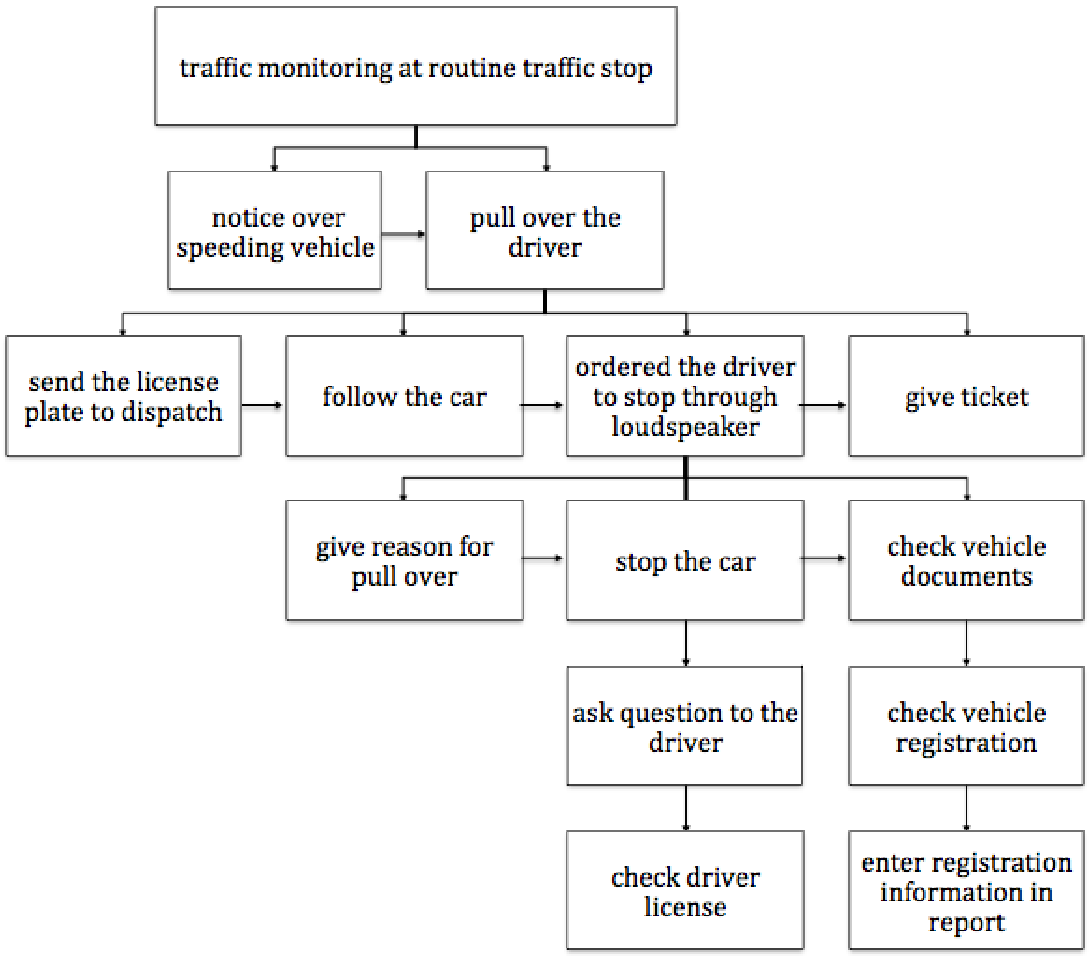
Smaller Story E6:



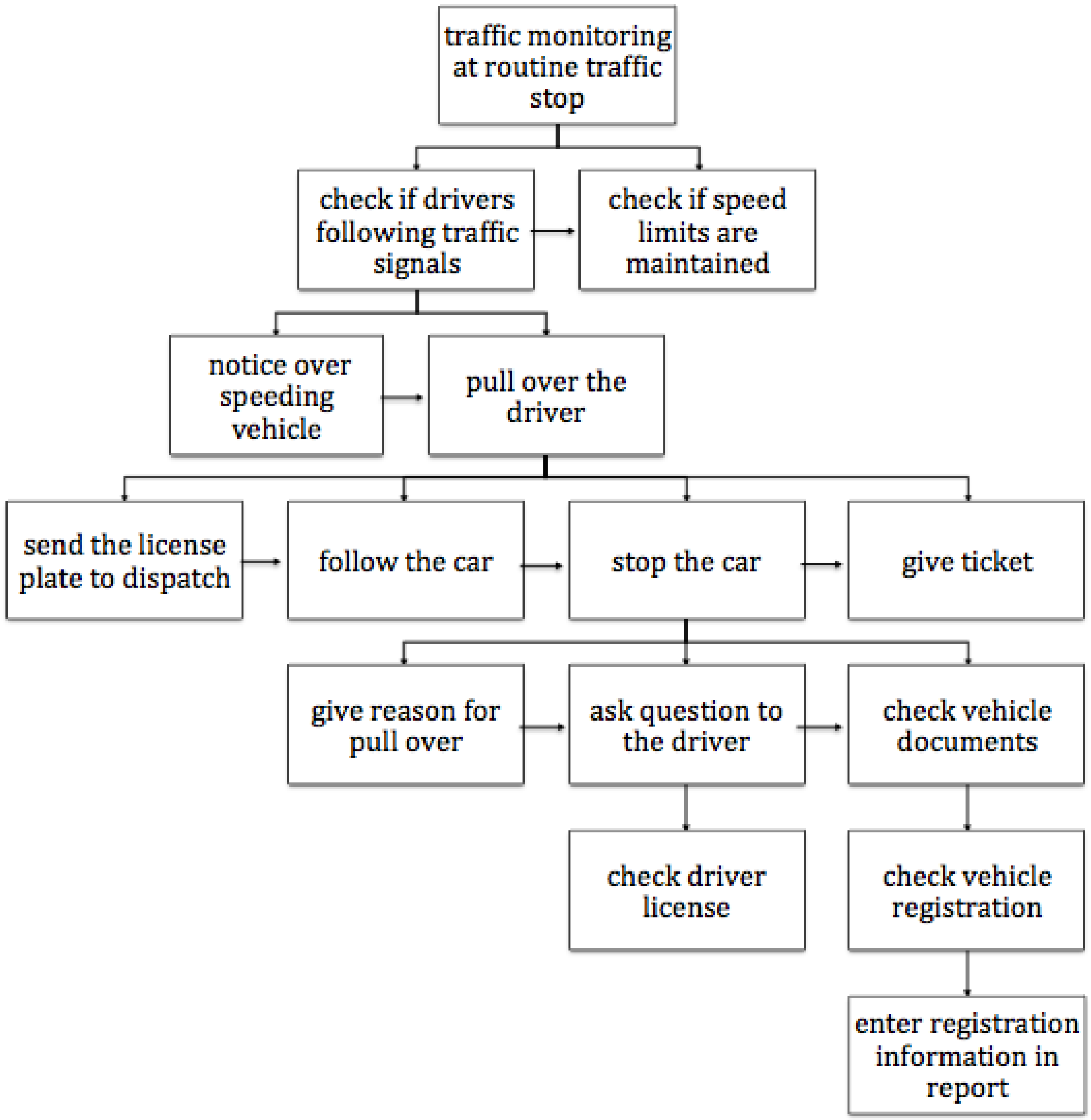
Smaller Story E7:



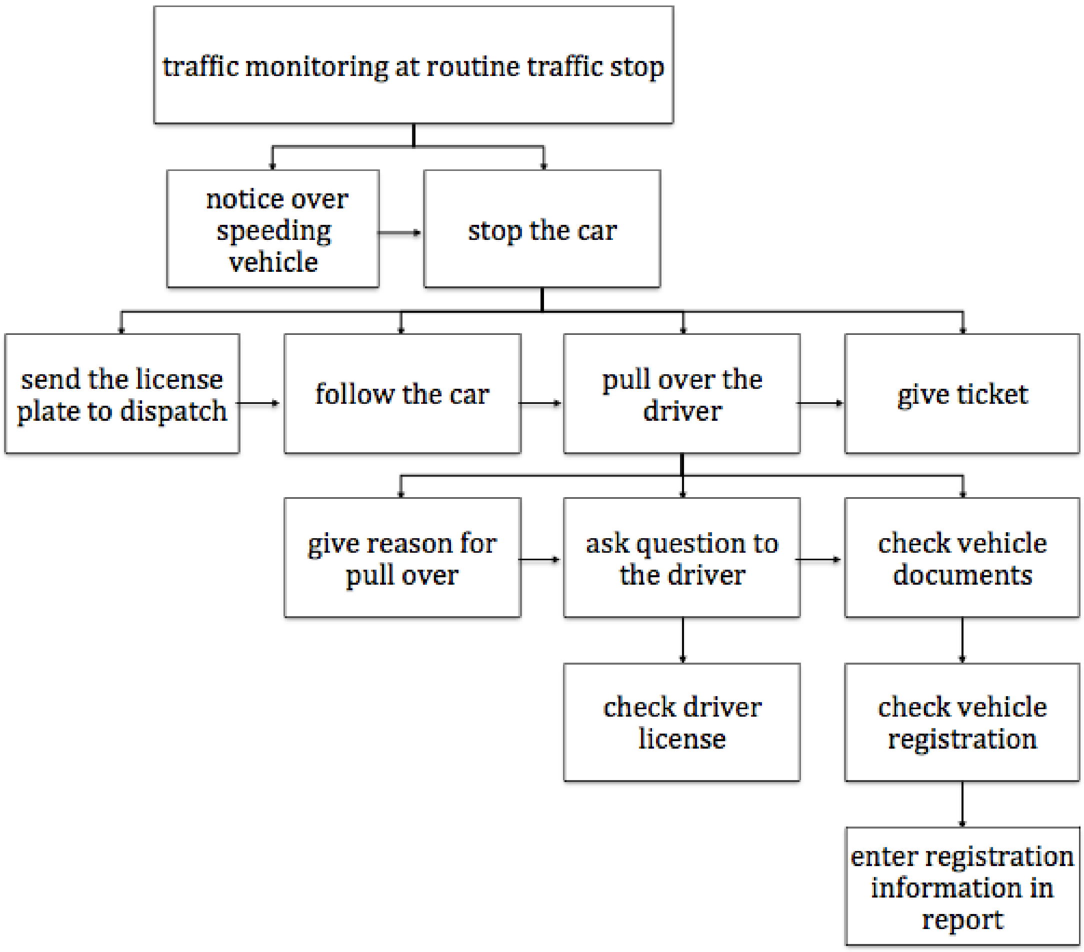
Smaller Story E8:



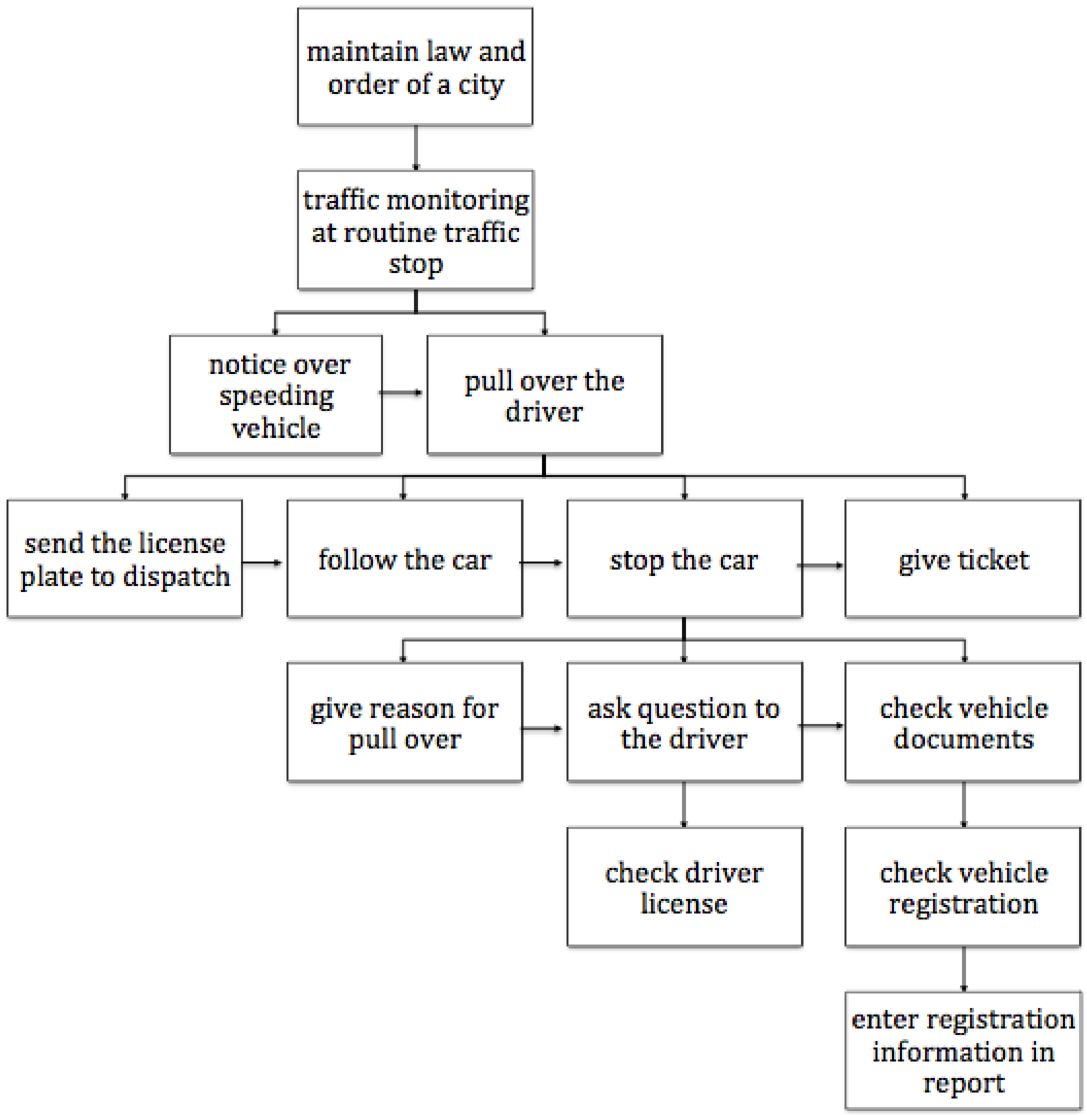
Smaller Story E9:



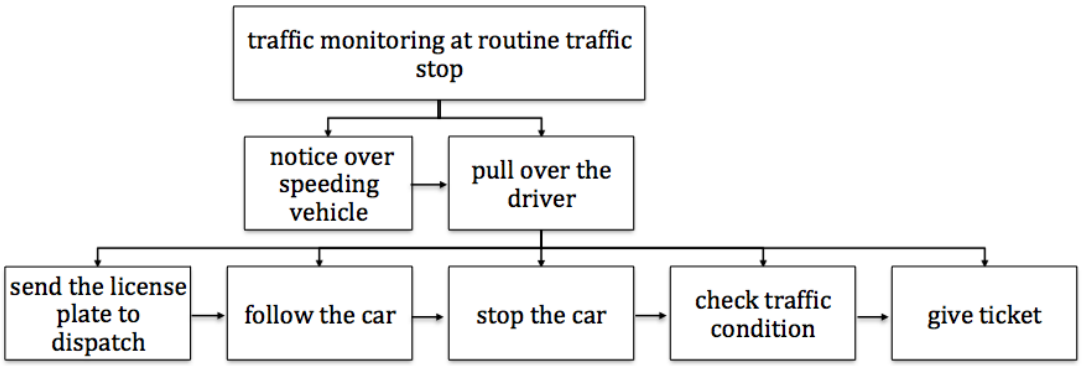
Smaller Story E10:



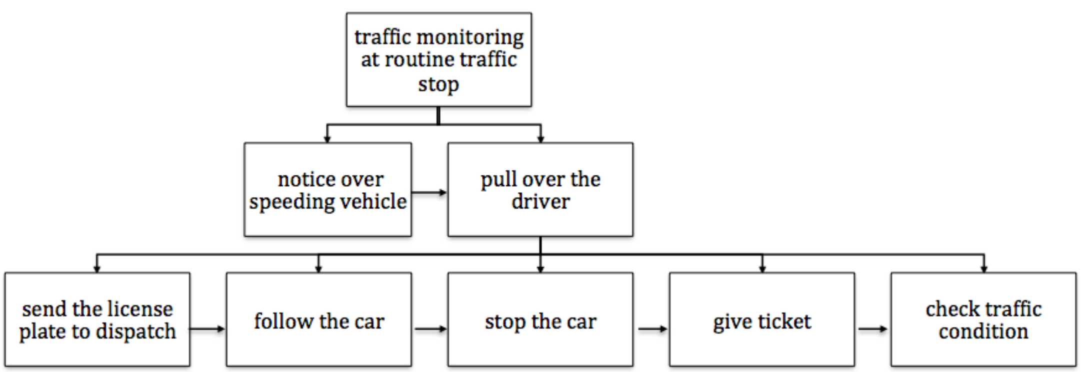
Smaller Story E11:



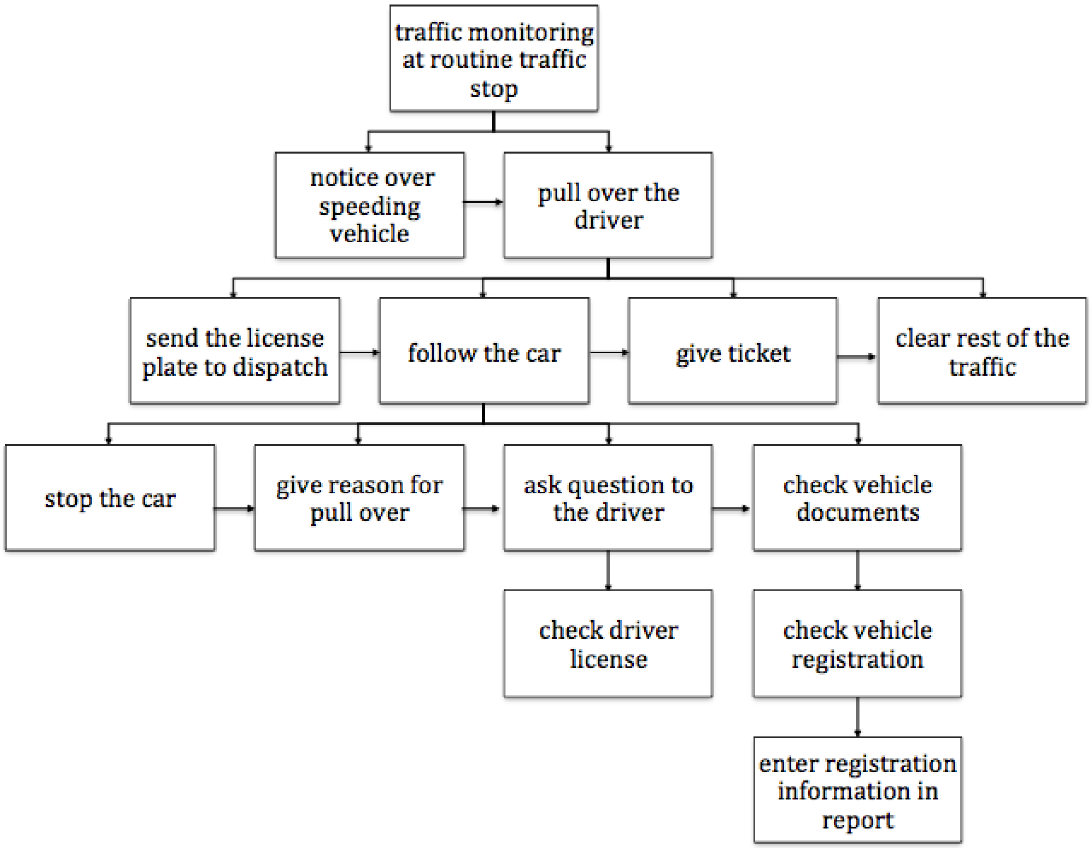
Smaller Story E12:



Smaller Story E13:



Smaller Story E14:



Appendix B

GXL REPRESENTATION OF SMALLER STORY GRAPHS

Smaller Story E1:


```
<?xml version="1.0"?>
<!DOCTYPE gxl SYSTEM "http://www.gupro.de/GXL/gxl-1.0.dtd">

<gxl><graph id="smallerE1" edgeids="false" edgemode="directed">

<node id="_1"><attr name="label"><string>traffic monitoring at routine traffic
stop</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_2"><attr name="label"><string>notice over speeding
vehicle</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_3"><attr name="label"><string>pull over the driver</string></attr><attr
name="type"><int>1</int></attr></node>
<node id="_4"><attr name="label"><string>send the license plate to
dispatch</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_5"><attr name="label"><string>follow the car</string></attr><attr
name="type"><int>2</int></attr></node>
<node id="_6"><attr name="label"><string>stop the car</string></attr><attr
name="type"><int>1</int></attr></node>
```

<node id="_7"><attr name="label"><string>give ticket</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_8"><attr name="label"><string>give reason for pull over</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_9"><attr name="label"><string>ask question to the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_10"><attr name="label"><string>check vehicle documents</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_24"><attr name="label"><string>check influence of alcohol</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_28"><attr name="label"><string>check possession of illegal arms</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_26"><attr name="label"><string>driver fails in sobriety test</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_30"><attr name="label"><string>find no arms</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_45"><attr name="label"><string>driver does not have vehicle registration documents</string></attr><attr name="type"><int>2</int></attr></node>

<edge from="_1" to="_2"><attr name="distance"><int>1</int></attr></edge>
<edge from="_1" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_2" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_4"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_5"><attr name="distance"><int>1</int></attr></edge>

<edge from="_3" to="_6"><attr name="distance"><int>1</int></attr></edge>

<edge from="_3" to="_7"><attr name="distance"><int>1</int></attr></edge>

<edge from="_4" to="_5"><attr name="distance"><int>1</int></attr></edge>

<edge from="_5" to="_6"><attr name="distance"><int>1</int></attr></edge>

<edge from="_6" to="_7"><attr name="distance"><int>1</int></attr></edge>

<edge from="_6" to="_8"><attr name="distance"><int>1</int></attr></edge>

<edge from="_6" to="_9"><attr name="distance"><int>1</int></attr></edge>

<edge from="_6" to="_10"><attr name="distance"><int>1</int></attr></edge>

<edge from="_8" to="_9"><attr name="distance"><int>1</int></attr></edge>

<edge from="_9" to="_10"><attr name="distance"><int>1</int></attr></edge>

<edge from="_9" to="_24"><attr name="distance"><int>1</int></attr></edge>

<edge from="_9" to="_28"><attr name="distance"><int>1</int></attr></edge>

<edge from="_24" to="_28"><attr name="distance"><int>1</int></attr></edge>

<edge from="_10" to="_45"><attr name="distance"><int>1</int></attr></edge>

<edge from="_24" to="_26"><attr name="distance"><int>1</int></attr></edge>

<edge from="_28" to="_30"><attr name="distance"><int>1</int></attr></edge>

</graph></gxl>

Smaller Story E2:

<?xml version="1.0"?>

<!DOCTYPE gxl SYSTEM "http://www.gupro.de/GXL/gxl-1.0.dtd">

<gxl><graph id="smallerE2" edgeids="false" edgemode="directed">

<node id="_1"><attr name="label"><string>traffic monitoring at routine traffic stop</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_2"><attr name="label"><string>notice over speeding vehicle</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_3"><attr name="label"><string>pull over the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_4"><attr name="label"><string>send the license plate to dispatch</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_5"><attr name="label"><string>follow the car</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_6"><attr name="label"><string>stop the car</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_7"><attr name="label"><string>give ticket</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_8"><attr name="label"><string>give reason for pull over</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_9"><attr name="label"><string>ask question to the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_10"><attr name="label"><string>check vehicle documents</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_13"><attr name="label"><string>check driver license</string></attr><attr name="type"><int>2</int></attr></node>

<node id="_14"><attr name="label"><string>check vehicle registration</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_15"><attr name="label"><string>check vehicle insurance status</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_20"><attr name="label"><string>enter registration information in report</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_21"><attr name="label"><string>find insurance is ok</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_22"><attr name="label"><string>enter insurance information in report</string></attr><attr name="type"><int>2</int></attr></node>

<edge from="_1" to="_2"><attr name="distance"><int>1</int></attr></edge>
<edge from="_1" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_2" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_4"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_4" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_5" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_8"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_9"><attr name="distance"><int>1</int></attr></edge>

<edge from="_6" to="_10"><attr name="distance"><int>1</int></attr></edge>
<edge from="_8" to="_9"><attr name="distance"><int>1</int></attr></edge>
<edge from="_9" to="_10"><attr name="distance"><int>1</int></attr></edge>
<edge from="_9" to="_13"><attr name="distance"><int>1</int></attr></edge>
<edge from="_10" to="_14"><attr name="distance"><int>1</int></attr></edge>
<edge from="_10" to="_15"><attr name="distance"><int>1</int></attr></edge>
<edge from="_14" to="_15"><attr name="distance"><int>1</int></attr></edge>
<edge from="_14" to="_20"><attr name="distance"><int>1</int></attr></edge>
<edge from="_15" to="_21"><attr name="distance"><int>1</int></attr></edge>
<edge from="_15" to="_22"><attr name="distance"><int>1</int></attr></edge>
<edge from="_21" to="_22"><attr name="distance"><int>1</int></attr></edge>

</graph></gxl>

<u>Smaller Story E3:</u>

<?xml version="1.0"?>
<!DOCTYPE gxl SYSTEM "http://www.gupro.de/GXL/gxl-1.0.dtd">

<gxl><graph id="smallerE3" edgeids="false" edgemode="directed">

<node id="_46"><attr name="label"><string>maintain law and order of a
city</string></attr><attr name="type"><int>1</int></attr></node>

<node id="_47"><attr name="label"><string>patrolling the roads</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_48"><attr name="label"><string>submitted report to the supervisor</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_1"><attr name="label"><string>traffic monitoring at routine traffic stop</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_2"><attr name="label"><string>notice over speeding vehicle</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_3"><attr name="label"><string>pull over the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_4"><attr name="label"><string>send the license plate to dispatch</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_5"><attr name="label"><string>follow the car</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_6"><attr name="label"><string>stop the car</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_7"><attr name="label"><string>give ticket</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_8"><attr name="label"><string>give reason for pull over</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_9"><attr name="label"><string>ask question to the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_10"><attr name="label"><string>check vehicle documents</string></attr><attr name="type"><int>1</int></attr></node>

<node id="_13"><attr name="label"><string>check driver license</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_14"><attr name="label"><string>check vehicle registration</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_20"><attr name="label"><string>enter registration information in report</string></attr><attr name="type"><int>2</int></attr></node>

<edge from="_46" to="_1"><attr name="distance"><int>1</int></attr></edge>
<edge from="_46" to="_47"><attr name="distance"><int>1</int></attr></edge>
<edge from="_46" to="_48"><attr name="distance"><int>1</int></attr></edge>
<edge from="_1" to="_47"><attr name="distance"><int>1</int></attr></edge>
<edge from="_47" to="_48"><attr name="distance"><int>1</int></attr></edge>
<edge from="_1" to="_2"><attr name="distance"><int>1</int></attr></edge>
<edge from="_1" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_2" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_4"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_4" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_5" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_8"><attr name="distance"><int>1</int></attr></edge>

<edge from="_6" to="_9"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_10"><attr name="distance"><int>1</int></attr></edge>
<edge from="_8" to="_9"><attr name="distance"><int>1</int></attr></edge>
<edge from="_9" to="_10"><attr name="distance"><int>1</int></attr></edge>
<edge from="_9" to="_13"><attr name="distance"><int>1</int></attr></edge>
<edge from="_10" to="_14"><attr name="distance"><int>1</int></attr></edge>
<edge from="_14" to="_20"><attr name="distance"><int>1</int></attr></edge>

</graph></gxl>

<u>Smaller Story E4:</u>

<?xml version="1.0"?>
<!DOCTYPE gxl SYSTEM "http://www.gupro.de/GXL/gxl-1.0.dtd">

<gxl><graph id="smallerE4" edgeids="false" edgemode="directed">

<node id="_1"><attr name="label"><string>traffic monitoring at routine traffic stop</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_2"><attr name="label"><string>notice over speeding vehicle</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_3"><attr name="label"><string>pull over the driver</string></attr><attr name="type"><int>1</int></attr></node>

<node id="_4"><attr name="label"><string>send the license plate to dispatch</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_5"><attr name="label"><string>follow the car</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_6"><attr name="label"><string>stop the car</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_49"><attr name="label"><string>check previous violation records</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_7"><attr name="label"><string>give ticket</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_8"><attr name="label"><string>give reason for pull over</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_9"><attr name="label"><string>ask question to the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_10"><attr name="label"><string>check vehicle documents</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_13"><attr name="label"><string>check driver license</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_14"><attr name="label"><string>check vehicle registration</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_20"><attr name="label"><string>enter registration information in report</string></attr><attr name="type"><int>2</int></attr></node>

<edge from="_1" to="_2"><attr name="distance"><int>1</int></attr></edge>

<edge from="_1" to="_3"><attr name="distance"><int>1</int></attr></edge>

<edge from="_2" to="_3"><attr name="distance"><int>1</int></attr></edge>

<edge from="_3" to="_4"><attr name="distance"><int>1</int></attr></edge>

<edge from="_3" to="_5"><attr name="distance"><int>1</int></attr></edge>

<edge from="_3" to="_6"><attr name="distance"><int>1</int></attr></edge>

<edge from="_3" to="_49"><attr name="distance"><int>1</int></attr></edge>

<edge from="_3" to="_7"><attr name="distance"><int>1</int></attr></edge>

<edge from="_4" to="_5"><attr name="distance"><int>1</int></attr></edge>

<edge from="_5" to="_6"><attr name="distance"><int>1</int></attr></edge>

<edge from="_6" to="_49"><attr name="distance"><int>1</int></attr></edge>

<edge from="_49" to="_7"><attr name="distance"><int>1</int></attr></edge>

<edge from="_6" to="_8"><attr name="distance"><int>1</int></attr></edge>

<edge from="_6" to="_9"><attr name="distance"><int>1</int></attr></edge>

<edge from="_6" to="_10"><attr name="distance"><int>1</int></attr></edge>

<edge from="_8" to="_9"><attr name="distance"><int>1</int></attr></edge>

<edge from="_9" to="_10"><attr name="distance"><int>1</int></attr></edge>

<edge from="_9" to="_13"><attr name="distance"><int>1</int></attr></edge>

<edge from="_10" to="_14"><attr name="distance"><int>1</int></attr></edge>

<edge from="_14" to="_20"><attr name="distance"><int>1</int></attr></edge>

</graph></gxl>

Smaller Story E5:

```
<?xml version="1.0"?>
<!DOCTYPE gxl SYSTEM "http://www.gupro.de/GXL/gxl-1.0.dtd">

<gxl><graph id="smallerE5" edgeids="false" edgemode="directed">

<node id="_50"><attr name="label"><string>maintain traffic in the
highway</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_2"><attr name="label"><string>notice over speeding
vehicle</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_3"><attr name="label"><string>pull over the driver</string></attr><attr
name="type"><int>1</int></attr></node>
<node id="_4"><attr name="label"><string>send the license plate to
dispatch</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_5"><attr name="label"><string>follow the car</string></attr><attr
name="type"><int>2</int></attr></node>
<node id="_6"><attr name="label"><string>stop the car</string></attr><attr
name="type"><int>1</int></attr></node>
<node id="_7"><attr name="label"><string>give ticket</string></attr><attr
name="type"><int>2</int></attr></node>
<node id="_8"><attr name="label"><string>give reason for pull
over</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_9"><attr name="label"><string>ask question to the
driver</string></attr><attr name="type"><int>1</int></attr></node>
```

<node id="_10"><attr name="label"><string>check vehicle documents</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_13"><attr name="label"><string>check driver license</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_14"><attr name="label"><string>check vehicle registration</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_20"><attr name="label"><string>enter registration information in report</string></attr><attr name="type"><int>2</int></attr></node>

<edge from="_50" to="_2"><attr name="distance"><int>1</int></attr></edge>
<edge from="_50" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_2" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_4"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_4" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_5" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_8"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_9"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_10"><attr name="distance"><int>1</int></attr></edge>
<edge from="_8" to="_9"><attr name="distance"><int>1</int></attr></edge>
<edge from="_9" to="_10"><attr name="distance"><int>1</int></attr></edge>

<edge from="_9" to="_13"><attr name="distance"><int>1</int></attr></edge>
<edge from="_10" to="_14"><attr name="distance"><int>1</int></attr></edge>
<edge from="_14" to="_20"><attr name="distance"><int>1</int></attr></edge>

</graph></gxl>

<u>Smaller Story E6:</u>

<?xml version="1.0"?>
<!DOCTYPE gxl SYSTEM "http://www.gupro.de/GXL/gxl-1.0.dtd">

<gxl><graph id="smallerE6" edgeids="false" edgemode="directed">

<node id="_1"><attr name="label"><string>traffic monitoring at routine traffic stop</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_2"><attr name="label"><string>notice over speeding vehicle</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_3"><attr name="label"><string>pull over the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_4"><attr name="label"><string>send the license plate to dispatch</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_5"><attr name="label"><string>follow the car</string></attr><attr name="type"><int>2</int></attr></node>

<node id="_51"><attr name="label"><string>ordered the driver to stop through loudspeaker</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_7"><attr name="label"><string>give ticket</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_8"><attr name="label"><string>give reason for pull over</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_9"><attr name="label"><string>ask question to the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_10"><attr name="label"><string>check vehicle documents</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_13"><attr name="label"><string>check driver license</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_14"><attr name="label"><string>check vehicle registration</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_20"><attr name="label"><string>enter registration information in report</string></attr><attr name="type"><int>2</int></attr></node>

<edge from="_1" to="_2"><attr name="distance"><int>1</int></attr></edge>
<edge from="_1" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_2" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_4"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_51"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_7"><attr name="distance"><int>1</int></attr></edge>

```
<edge from="_4" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_5" to="_51"><attr name="distance"><int>1</int></attr></edge>
<edge from="_51" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_51" to="_8"><attr name="distance"><int>1</int></attr></edge>
<edge from="_51" to="_9"><attr name="distance"><int>1</int></attr></edge>
<edge from="_51" to="_10"><attr name="distance"><int>1</int></attr></edge>
<edge from="_8" to="_9"><attr name="distance"><int>1</int></attr></edge>
<edge from="_9" to="_10"><attr name="distance"><int>1</int></attr></edge>
<edge from="_9" to="_13"><attr name="distance"><int>1</int></attr></edge>
<edge from="_10" to="_14"><attr name="distance"><int>1</int></attr></edge>
<edge from="_14" to="_20"><attr name="distance"><int>1</int></attr></edge>

</graph></gxl>
```


Smaller Story E7:


```
<?xml version="1.0"?>
<!DOCTYPE gxl SYSTEM "http://www.gupro.de/GXL/gxl-1.0.dtd">

<gxl><graph id="smallerE7" edgeids="false" edgemode="directed">

<node id="_1"><attr name="label"><string>traffic monitoring at routine traffic
stop</string></attr><attr name="type"><int>1</int></attr></node>
```

<node id="_2"><attr name="label"><string>notice over speeding vehicle</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_3"><attr name="label"><string>pull over the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_4"><attr name="label"><string>send the license plate to dispatch</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_5"><attr name="label"><string>follow the car</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_6"><attr name="label"><string>stop the car</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_7"><attr name="label"><string>give ticket</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_8"><attr name="label"><string>give reason for pull over</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_9"><attr name="label"><string>ask question to the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_25"><attr name="label"><string>take sobriety test of driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_26"><attr name="label"><string>driver fails in sobriety test</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_10"><attr name="label"><string>check vehicle documents</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_13"><attr name="label"><string>check driver license</string></attr><attr name="type"><int>2</int></attr></node>

<node id="_14"><attr name="label"><string>check vehicle
registration</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_20"><attr name="label"><string>enter registration information in
report</string></attr><attr name="type"><int>2</int></attr></node>

<edge from="_1" to="_2"><attr name="distance"><int>1</int></attr></edge>
<edge from="_1" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_2" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_4"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_4" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_5" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_8"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_9"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_10"><attr name="distance"><int>1</int></attr></edge>
<edge from="_8" to="_9"><attr name="distance"><int>1</int></attr></edge>
<edge from="_9" to="_10"><attr name="distance"><int>1</int></attr></edge>
<edge from="_9" to="_25"><attr name="distance"><int>1</int></attr></edge>
<edge from="_25" to="_26"><attr name="distance"><int>1</int></attr></edge>
<edge from="_26" to="_13"><attr name="distance"><int>1</int></attr></edge>
<edge from="_10" to="_14"><attr name="distance"><int>1</int></attr></edge>

<edge from="_14" to="_20"><attr name="distance"><int>1</int></attr></edge>

</graph></gxl>

<u>Smaller Story E8:</u>

<?xml version="1.0"?>
<!DOCTYPE gxl SYSTEM "http://www.gupro.de/GXL/gxl-1.0.dtd">

<gxl><graph id="smallerE8" edgeids="false" edgemode="directed">

<node id="_1"><attr name="label"><string>traffic monitoring at routine traffic stop</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_2"><attr name="label"><string>notice over speeding vehicle</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_3"><attr name="label"><string>pull over the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_4"><attr name="label"><string>send the license plate to dispatch</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_5"><attr name="label"><string>follow the car</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_51"><attr name="label"><string>ordered the driver to stop through loudspeaker</string></attr><attr name="type"><int>1</int></attr></node>

<node id="_6"><attr name="label"><string>stop the car</string></attr><attr name="type"><int>1</int></attr></node>

<node id="_7"><attr name="label"><string>give ticket</string></attr><attr name="type"><int>2</int></attr></node>

<node id="_8"><attr name="label"><string>give reason for pull over</string></attr><attr name="type"><int>2</int></attr></node>

<node id="_9"><attr name="label"><string>ask question to the driver</string></attr><attr name="type"><int>1</int></attr></node>

<node id="_10"><attr name="label"><string>check vehicle documents</string></attr><attr name="type"><int>1</int></attr></node>

<node id="_13"><attr name="label"><string>check driver license</string></attr><attr name="type"><int>2</int></attr></node>

<node id="_14"><attr name="label"><string>check vehicle registration</string></attr><attr name="type"><int>1</int></attr></node>

<node id="_20"><attr name="label"><string>enter registration information in report</string></attr><attr name="type"><int>2</int></attr></node>

<edge from="_1" to="_2"><attr name="distance"><int>1</int></attr></edge>

<edge from="_1" to="_3"><attr name="distance"><int>1</int></attr></edge>

<edge from="_2" to="_3"><attr name="distance"><int>1</int></attr></edge>

<edge from="_3" to="_4"><attr name="distance"><int>1</int></attr></edge>

<edge from="_3" to="_5"><attr name="distance"><int>1</int></attr></edge>

<edge from="_3" to="_51"><attr name="distance"><int>1</int></attr></edge>

<edge from="_3" to="_7"><attr name="distance"><int>1</int></attr></edge>

<edge from="_4" to="_5"><attr name="distance"><int>1</int></attr></edge>

<edge from="_5" to="_51"><attr name="distance"><int>1</int></attr></edge>

<edge from="_51" to="_7"><attr name="distance"><int>1</int></attr></edge>

<edge from="_51" to="_8"><attr name="distance"><int>1</int></attr></edge>

<edge from="_51" to="_6"><attr name="distance"><int>1</int></attr></edge>

<edge from="_6" to="_10"><attr name="distance"><int>1</int></attr></edge>

<edge from="_8" to="_6"><attr name="distance"><int>1</int></attr></edge>

<edge from="_6" to="_10"><attr name="distance"><int>1</int></attr></edge>

<edge from="_6" to="_9"><attr name="distance"><int>1</int></attr></edge>

<edge from="_9" to="_13"><attr name="distance"><int>1</int></attr></edge>

<edge from="_10" to="_14"><attr name="distance"><int>1</int></attr></edge>

<edge from="_14" to="_20"><attr name="distance"><int>1</int></attr></edge>

</graph></gxl>

<u>Smaller Story E9:</u>

<?xml version="1.0"?>
<!DOCTYPE gxl SYSTEM "http://www.gupro.de/GXL/gxl-1.0.dtd">

<gxl><graph id="smallerE9" edgeids="false" edgemode="directed">

<node id="_1"><attr name="label"><string>traffic monitoring at routine traffic stop</string></attr><attr name="type"><int>1</int></attr></node>

<node id="_55"><attr name="label"><string>check if drivers following traffic signals</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_52"><attr name="label"><string>check if speed limits are maintained</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_2"><attr name="label"><string>notice over speeding vehicle</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_3"><attr name="label"><string>pull over the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_4"><attr name="label"><string>send the license plate to dispatch</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_5"><attr name="label"><string>follow the car</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_6"><attr name="label"><string>stop the car</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_7"><attr name="label"><string>give ticket</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_8"><attr name="label"><string>give reason for pull over</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_9"><attr name="label"><string>ask question to the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_10"><attr name="label"><string>check vehicle documents</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_13"><attr name="label"><string>check driver license</string></attr><attr name="type"><int>2</int></attr></node>

<node id="_14"><attr name="label"><string>check vehicle
registration</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_20"><attr name="label"><string>enter registration information in
report</string></attr><attr name="type"><int>2</int></attr></node>

<edge from="_1" to="_55"><attr name="distance"><int>1</int></attr></edge>
<edge from="_1" to="_52"><attr name="distance"><int>1</int></attr></edge>
<edge from="_55" to="_52"><attr name="distance"><int>1</int></attr></edge>
<edge from="_55" to="_2"><attr name="distance"><int>1</int></attr></edge>
<edge from="_55" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_2" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_4"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_4" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_5" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_8"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_9"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_10"><attr name="distance"><int>1</int></attr></edge>
<edge from="_8" to="_9"><attr name="distance"><int>1</int></attr></edge>
<edge from="_9" to="_10"><attr name="distance"><int>1</int></attr></edge>
<edge from="_9" to="_13"><attr name="distance"><int>1</int></attr></edge>

<edge from="_10" to="_14"><attr name="distance"><int>1</int></attr></edge>
<edge from="_14" to="_20"><attr name="distance"><int>1</int></attr></edge>

</graph></gxl>

<u>Smaller Story E10:</u>

<?xml version="1.0"?>
<!DOCTYPE gxl SYSTEM "http://www.gupro.de/GXL/gxl-1.0.dtd">

<gxl><graph id="smallerE10" edgeids="false" edgemode="directed">

<node id="_1"><attr name="label"><string>traffic monitoring at routine traffic stop</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_2"><attr name="label"><string>notice over speeding vehicle</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_3"><attr name="label"><string>pull over the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_4"><attr name="label"><string>send the license plate to dispatch</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_5"><attr name="label"><string>follow the car</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_6"><attr name="label"><string>stop the car</string></attr><attr name="type"><int>1</int></attr></node>

<node id="_7"><attr name="label"><string>give ticket</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_8"><attr name="label"><string>give reason for pull over</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_9"><attr name="label"><string>ask question to the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_10"><attr name="label"><string>check vehicle documents</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_13"><attr name="label"><string>check driver license</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_14"><attr name="label"><string>check vehicle registration</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_20"><attr name="label"><string>enter registration information in report</string></attr><attr name="type"><int>2</int></attr></node>

<edge from="_1" to="_2"><attr name="distance"><int>1</int></attr></edge>
<edge from="_1" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_2" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_4"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_4" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_5" to="_3"><attr name="distance"><int>1</int></attr></edge>

<edge from="_3" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_8"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_9"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_10"><attr name="distance"><int>1</int></attr></edge>
<edge from="_8" to="_9"><attr name="distance"><int>1</int></attr></edge>
<edge from="_9" to="_10"><attr name="distance"><int>1</int></attr></edge>
<edge from="_9" to="_13"><attr name="distance"><int>1</int></attr></edge>
<edge from="_10" to="_14"><attr name="distance"><int>1</int></attr></edge>
<edge from="_14" to="_20"><attr name="distance"><int>1</int></attr></edge>

</graph></gxl>

Smaller Story E11:

<?xml version="1.0"?>
<!DOCTYPE gxl SYSTEM "http://www.gupro.de/GXL/gxl-1.0.dtd">

<gxl><graph id="smallerE11" edgeids="false" edgemode="directed">

<node id="_46"><attr name="label"><string>maintain law and order of a city</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_1"><attr name="label"><string>traffic monitoring at routine traffic stop</string></attr><attr name="type"><int>1</int></attr></node>

<node id="_2"><attr name="label"><string>notice over speeding vehicle</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_3"><attr name="label"><string>pull over the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_4"><attr name="label"><string>send the license plate to dispatch</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_5"><attr name="label"><string>follow the car</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_6"><attr name="label"><string>stop the car</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_7"><attr name="label"><string>give ticket</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_8"><attr name="label"><string>give reason for pull over</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_9"><attr name="label"><string>ask question to the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_10"><attr name="label"><string>check vehicle documents</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_13"><attr name="label"><string>check driver license</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_14"><attr name="label"><string>check vehicle registration</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_20"><attr name="label"><string>enter registration information in report</string></attr><attr name="type"><int>2</int></attr></node>

<edge from="_46" to="_1"><attr name="distance"><int>1</int></attr></edge>
<edge from="_1" to="_2"><attr name="distance"><int>1</int></attr></edge>
<edge from="_1" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_2" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_4"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_4" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_5" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_8"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_9"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_10"><attr name="distance"><int>1</int></attr></edge>
<edge from="_8" to="_9"><attr name="distance"><int>1</int></attr></edge>
<edge from="_9" to="_10"><attr name="distance"><int>1</int></attr></edge>
<edge from="_9" to="_13"><attr name="distance"><int>1</int></attr></edge>
<edge from="_10" to="_14"><attr name="distance"><int>1</int></attr></edge>
<edge from="_14" to="_20"><attr name="distance"><int>1</int></attr></edge>

</graph></gxl>

Smaller Story E12:


```
<?xml version="1.0"?>
<!DOCTYPE gxl SYSTEM "http://www.gupro.de/GXL/gxl-1.0.dtd">

<gxl><graph id="smallerE12" edgeids="false" edgemode="directed">

<node id="_1"><attr name="label"><string>traffic monitoring at routine traffic
stop</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_2"><attr name="label"><string>notice over speeding
vehicle</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_3"><attr name="label"><string>pull over the driver</string></attr><attr
name="type"><int>1</int></attr></node>
<node id="_4"><attr name="label"><string>send the license plate to
dispatch</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_5"><attr name="label"><string>follow the car</string></attr><attr
name="type"><int>2</int></attr></node>
<node id="_6"><attr name="label"><string>stop the car</string></attr><attr
name="type"><int>2</int></attr></node>
<node id="_7"><attr name="label"><string>give ticket</string></attr><attr
name="type"><int>2</int></attr></node>
<node id="_11"><attr name="label"><string>check the traffic
condition</string></attr><attr name="type"><int>2</int></attr></node>
```

```
<edge from="_1" to="_2"><attr name="distance"><int>1</int></attr></edge>
<edge from="_1" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_2" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_4"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_11"><attr name="distance"><int>1</int></attr></edge>
<edge from="_4" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_5" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_11"><attr name="distance"><int>1</int></attr></edge>
<edge from="_11" to="_7"><attr name="distance"><int>1</int></attr></edge>

</graph></gxl>
```


Smaller Story E13:


```
<?xml version="1.0"?>
<!DOCTYPE gxl SYSTEM "http://www.gupro.de/GXL/gxl-1.0.dtd">

<gxl><graph id="smallerE13" edgeids="false" edgemode="directed">

<node id="_1"><attr name="label"><string>traffic monitoring at routine traffic
stop</string></attr><attr name="type"><int>1</int></attr></node>
```

<node id="_2"><attr name="label"><string>notice over speeding vehicle</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_3"><attr name="label"><string>pull over the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_4"><attr name="label"><string>send the license plate to dispatch</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_5"><attr name="label"><string>follow the car</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_6"><attr name="label"><string>stop the car</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_7"><attr name="label"><string>give ticket</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_11"><attr name="label"><string>check the traffic condition</string></attr><attr name="type"><int>2</int></attr></node>

<edge from="_1" to="_2"><attr name="distance"><int>1</int></attr></edge>
<edge from="_1" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_2" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_4"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_5"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_11"><attr name="distance"><int>1</int></attr></edge>
<edge from="_4" to="_5"><attr name="distance"><int>1</int></attr></edge>

<edge from="_5" to="_6"><attr name="distance"><int>1</int></attr></edge>
<edge from="_6" to="_7"><attr name="distance"><int>1</int></attr></edge>
<edge from="_7" to="_11"><attr name="distance"><int>1</int></attr></edge>

</graph></gxl>

<u>Smaller Story E14:</u>

<?xml version="1.0"?>
<!DOCTYPE gxl SYSTEM "http://www.gupro.de/GXL/gxl-1.0.dtd">

<gxl><graph id="smallerE14" edgeids="false" edgemode="directed">

<node id="_1"><attr name="label"><string>traffic monitoring at routine traffic stop</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_2"><attr name="label"><string>notice over speeding vehicle</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_3"><attr name="label"><string>pull over the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_4"><attr name="label"><string>send the license plate to dispatch</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_5"><attr name="label"><string>follow the car</string></attr><attr name="type"><int>1</int></attr></node>

<node id="_6"><attr name="label"><string>stop the car</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_7"><attr name="label"><string>give ticket</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_53"><attr name="label"><string>clear rest of the traffic</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_8"><attr name="label"><string>give reason for pull over</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_9"><attr name="label"><string>ask question to the driver</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_10"><attr name="label"><string>check vehicle documents</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_13"><attr name="label"><string>check driver license</string></attr><attr name="type"><int>2</int></attr></node>
<node id="_14"><attr name="label"><string>check vehicle registration</string></attr><attr name="type"><int>1</int></attr></node>
<node id="_20"><attr name="label"><string>enter registration information in report</string></attr><attr name="type"><int>2</int></attr></node>

<edge from="_1" to="_2"><attr name="distance"><int>1</int></attr></edge>
<edge from="_1" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_2" to="_3"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_4"><attr name="distance"><int>1</int></attr></edge>
<edge from="_3" to="_5"><attr name="distance"><int>1</int></attr></edge>

<edge from="_3" to="_53"><attr name="distance"><int>1</int></attr></edge>

<edge from="_3" to="_7"><attr name="distance"><int>1</int></attr></edge>

<edge from="_4" to="_5"><attr name="distance"><int>1</int></attr></edge>

<edge from="_5" to="_7"><attr name="distance"><int>1</int></attr></edge>

<edge from="_7" to="_53"><attr name="distance"><int>1</int></attr></edge>

<edge from="_5" to="_6"><attr name="distance"><int>1</int></attr></edge>

<edge from="_5" to="_8"><attr name="distance"><int>1</int></attr></edge>

<edge from="_5" to="_9"><attr name="distance"><int>1</int></attr></edge>

<edge from="_5" to="_10"><attr name="distance"><int>1</int></attr></edge>

<edge from="_6" to="_8"><attr name="distance"><int>1</int></attr></edge>

<edge from="_8" to="_9"><attr name="distance"><int>1</int></attr></edge>

<edge from="_9" to="_10"><attr name="distance"><int>1</int></attr></edge>

<edge from="_9" to="_13"><attr name="distance"><int>1</int></attr></edge>

<edge from="_10" to="_14"><attr name="distance"><int>1</int></attr></edge>

<edge from="_14" to="_20"><attr name="distance"><int>1</int></attr></edge>

</graph></gxl>

Appendix C

IRB APPROVAL LETTER

Below please find the IRB approval letter obtained from the research office of University of Delaware.

UNIVERSITY OF DELAWARE

RESEARCH OFFICE

210 Hullihen Hall
University of Delaware
Newark, Delaware 19716-1551
Ph: 302/831-2136
Fax: 302/831-2828

DATE:	July 11, 2011
TO:	Kristina Winbladh, PhD
FROM:	University of Delaware IRB
STUDY TITLE:	[254444-1] iMuse: Interactive Model-based Use-case and Storytelling Environment
SUBMISSION TYPE:	New Project
ACTION:	APPROVED
APPROVAL DATE:	July 11, 2011
EXPIRATION DATE:	July 10, 2012
REVIEW TYPE:	Expedited Review
REVIEW CATEGORY:	Expedited review category # 7

Thank you for your submission of New Project materials for this research study. The University of Delaware IRB has APPROVED your submission. This approval is based on an appropriate risk/benefit ratio and a study design wherein the risks have been minimized. All research must be conducted in accordance with this approved submission.

This submission has received Expedited Review based on the applicable federal regulation.

Please remember that informed consent is a process beginning with a description of the study and insurance of participant understanding followed by a signed consent form. Informed consent must continue throughout the study via a dialogue between the researcher and research participant. Federal regulations require each participant receive a copy of the signed consent document.

Please note that any revision to previously approved materials must be approved by this office prior to initiation. Please use the appropriate revision forms for this procedure.

All SERIOUS and UNEXPECTED adverse events must be reported to this office. Please use the appropriate adverse event forms for this procedure. All sponsor reporting requirements should also be followed.

Please report all NON-COMPLIANCE issues or COMPLAINTS regarding this study to this office.

Please note that all research records must be retained for a minimum of three years.

Based on the risks, this project requires Continuing Review by this office on an annual basis. Please use the appropriate renewal forms for this procedure.

If you have any questions, please contact Jody-Lynn Berg at (302) 831-1119 or jlberg@udel.edu. Please include your study title and reference number in all correspondence with this office.