

Universidade Federal do Espírito Santo

# A narratology meta-model in Ecore with online model visualization

Model Driven Development project report

Author: Bernardo Ferreira Bastos Braga

Teacher: João Paulo Andrade Almeida

March of 2014 - Vitória, Espírito Santo, Brazil

# 1.Introduction

*All media as extensions of ourselves serve to provide new transforming vision and awareness.*

-Marshall McLuhan [McLuhan, 1964]

This project, aligned with my (in development) masters dissertation, has the objective of employing narratives as aid to transfer knowledge encrypted in OntoUML conceptual models. These narratives serve as a common ground for discussing conceptual models, enhancing conversations between human beings. They can, for example, function as documentation, helping people study conceptual models or as mediators in validation sessions with business experts. Following that narratives aid human beings to understand, express information and learn<sup>1</sup> we hypothesize that they also help people interacting with formal models to understand, express and learn them. Storytelling has already been applied in business and it was found to help communication and agreement<sup>2</sup>.

Narratives, in this context, are used as tools for thinking, helping the challenging task of conceptual modelling. They are means for a modeller to understand how modelling decisions affect creation, change and destruction of objects. We argue that even modelling experts have difficulty to understand in totality the implications of their modelling choices. These narratives may help them understand the possible worlds prescribed in a meta-model. This is particularly useful if there are people involved who are not familiar with the modelling language at all.

The system developed here is based on EMF (Eclipse Modelling Framework) technologies such as Ecore and EMF-REST. It currently consists of three core steps:

1. Modelling stories in the Reflective Ecore Model Editor in xmi format
2. Serving these stories in a REST API using EMF-REST<sup>3</sup>
3. Consuming the stories in a web app for data visualization

The resulting system reported here is only the first step towards our goal. Nevertheless it serves as a prototype for a larger (future) app that may better serve the theory proposed in the dissertation. This larger app is designed to be a social network for sharing these narrative models. Along with story editing facilities, the social interactivity of liking, commenting, sharing or forking posts are contemporary digital media conventions that promote participation and content creation. We hope to create an environment where people may collectively create meaning and agreement over conceptual models. The model was developed with this ultimate app in mind, although the end application does not use all of it yet.

The remainder of this report is organized as follows: section 2 presents and justifies the narrative meta-model developed in Ecore; section 3 exemplifies the modelling of a story using the meta-model presented previously; section 4 presents the technological arrangement used to

---

<sup>1</sup> refer to [http://en.wikipedia.org/wiki/Storytelling#Storytelling\\_and\\_learning](http://en.wikipedia.org/wiki/Storytelling#Storytelling_and_learning) for an introduction on the relationship between storytelling and learning

<sup>2</sup> check [http://en.wikipedia.org/wiki/Storytelling#In\\_business](http://en.wikipedia.org/wiki/Storytelling#In_business) for examples

<sup>3</sup> <http://emf-rest.com/>

serve the models via REST API; section 5 presents a graphical interface for the interaction of stories and section 6 presents conclusions and future work.

## 2. Story Metamodelling

*“Very young children love and demand stories, and can understand complex matters presented as stories, when their powers of comprehending general concepts, paradigms, are almost nonexistent.”*

Oliver Sacks, [Sacks, 98]

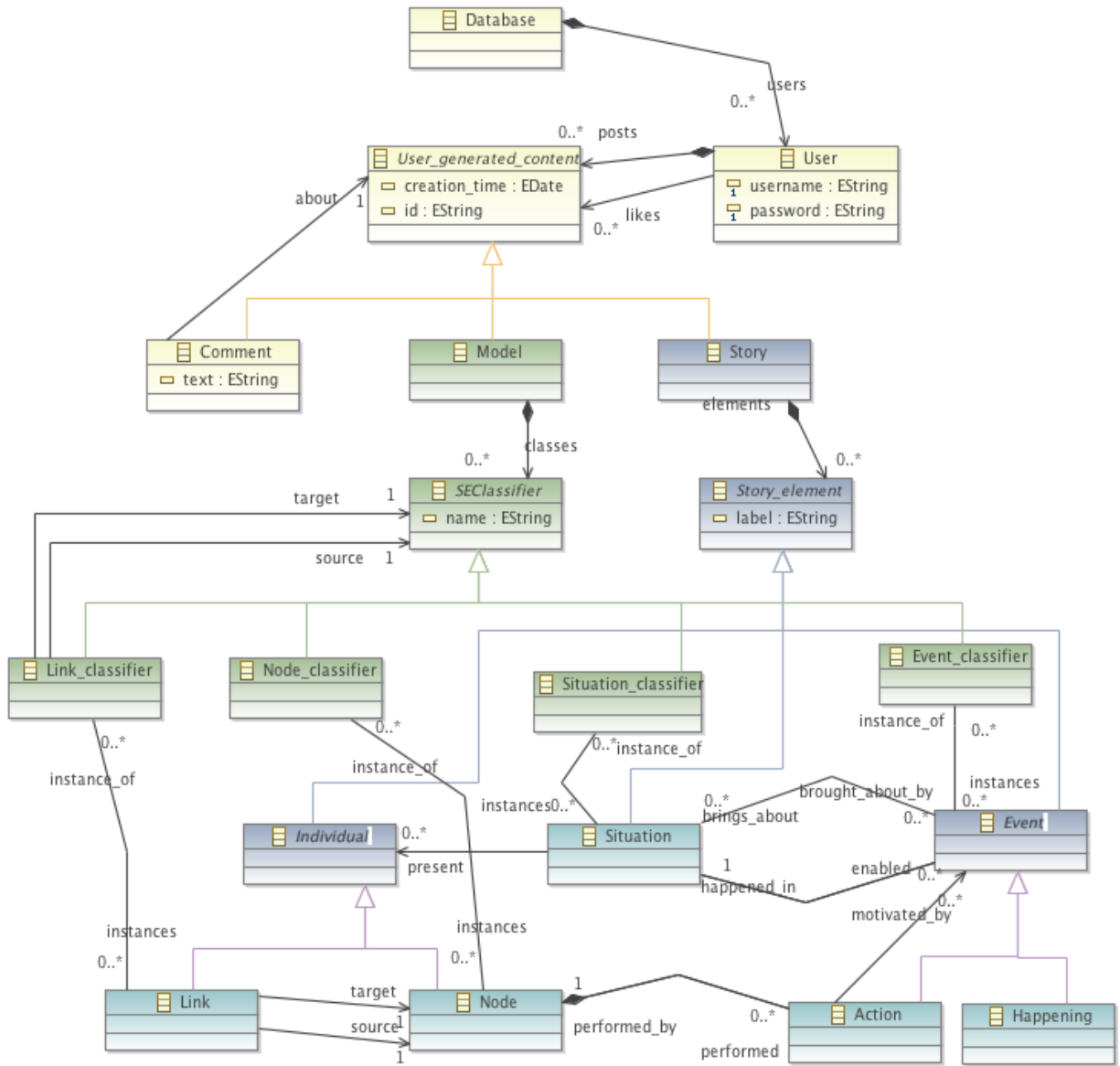
Our meta-model<sup>4</sup> for narratives was created using Ecore. It presumes the social network scenario presented in the first section, including concepts such as User Generated Content, Post, Comment etc.. The [model](#) is available at github in the EMF-REST components folder.

Although these concepts are important to characterize the media in which these narratives are told, the most elaborate concepts are about the content of this media, namely, the stories. A Story, in our model, is composed of Nodes (i.e. objects and agents), Links (i.e. relationships between nodes), Events (i.e. both Actions done by Agents and Happenings) and Situations (i.e. states of affairs that Nodes and Links may be present, brought about by events and that may enable other events). Our stories are essentially sequences of Situations connected by Events. In each situation there are Nodes and Links present. Each of the story elements may be classified according to a model. Therefore, each type of story element has an associated type of classifier. References to the model should be done in terms of the [OntoUML reference meta-model](#), but due to compatibility issues with EMF-REST, I've chosen to simplify this part of the model to release the first prototype. After dealing with the incompatibilities, effort was directed at providing new features for the web app. Future releases of the system should reference the existing OntoUML meta-model.

Here we refer to a narrative as the product of a narration and to stories as the framework that organizes the concepts of the narrative. A story may be told many times, each a different narrative of the same story. Therefore, as you interact with the web app presented in section 5 you are interacting with a narrative. The XMI model used for the example in section 3 (which in turn feeds the web app) would then be a formalized story. In other words, stories are instance-level models while narratives are manifestations of these stories.

---

<sup>4</sup> it should be noted that the model presented here is preliminary and should go through modifications in the near future.



### 3. Modelling a story (proof of concept)

*"Try not. Do... or do not. There is no try."*  
Yoda, [Star Wars, 77]

The meta-model described in the previous section may be better understood through an example. Here we will use Star Wars as our example story as it fits the structure of the Hero's Journey as described by Joseph Campbell [CAMPBELL, 93]. The story structure follows the diagram below.

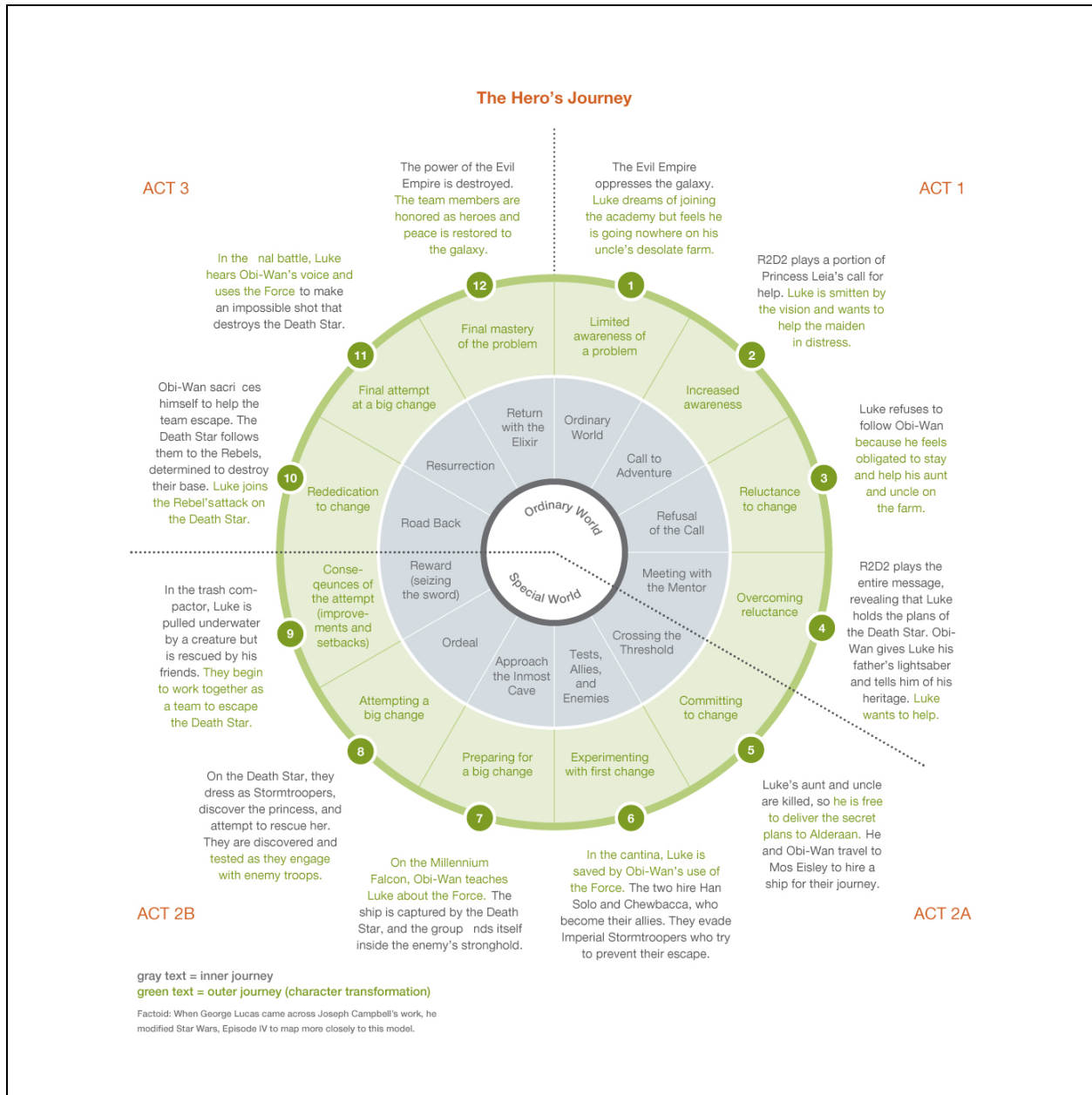


Figure 1: The Hero's Journey pattern applied to Star Wars  
available at [http://cdn.www.duarte.com/wp-content/uploads/2013/04/p33\\_hero-journey.jpg](http://cdn.www.duarte.com/wp-content/uploads/2013/04/p33_hero-journey.jpg)

In attachment 1 you will find the XMI that describes the first four situations of the Journey. I have interpreted the text describing each step of the journey and modeled the characters and their relationships, the situations and the events. These are, in turn, classified by some models, such as a model for Locations and a model for Genealogy.

In the first situation, there is Luke and he is at Luke's Uncle's Farm. Both Luke and the farm are nodes and "at" is a relationship between Luke and the farm. Luke is classified on the Genealogy model as a Person and the Farm as a Location in the location model, which also classifies the "at" association.

The next situation is enabled when R2D2 plays the cropped message to Luke (an Action), making him travel to find Obi Wan Kenobi.

The third situation is brought about by R2D2's action of playing the cropped message to Obi Wan. This action motivates Obi Wan to call Luke to follow him who, in turn, refuses Obi Wan (another action).

It should be noted that the model used in this Story is incomplete and some elements of the story are not classified at all. This is intentional, to make the process dynamic and sketch-like. The stories should be used to quickly express a point of view. If ontological concerns should be clarified, that could be done at a later stage, referencing a better model and classifying instances in order to clarify the semantics of the story. The point here is to leave the representation as simple as possible and leave blank the unthought-of aspects of the conceptualization.

The table below indicates which nodes and links occur in each situation and how events change situations. Relationships are indicated as "relationship name"<"source", "target">.

Star Wars: Storyline

Situations	Individuals	brought about by (event)	enabled (event)
Limited awareness of the problem	<ul style="list-style-type: none"> <li>• Luke,</li> <li>• Luke's Farm,</li> <li>• at&lt;Luke,Luke's Uncle's Farm&gt;</li> </ul>		
Increased Awareness	<ul style="list-style-type: none"> <li>• Luke</li> <li>• R2D2</li> </ul>	<ul style="list-style-type: none"> <li>• R2D2 plays Leia's message cropped to Luke</li> </ul>	<ul style="list-style-type: none"> <li>• Luke travels to find Obi Wan Kenobi</li> </ul>
Reluctance to change	<ul style="list-style-type: none"> <li>• Luke</li> <li>• R2D2</li> <li>• Obi Wan Kenobi</li> </ul>	<ul style="list-style-type: none"> <li>• R2D2 plays Leia's message cropped to Obi Wan Kenobi</li> </ul>	<ul style="list-style-type: none"> <li>• Luke refuses Obi Wan Kenobi</li> </ul>

		<ul style="list-style-type: none"><li>• Obi Wan Kenobi asks Luke to join him</li></ul>	
Overcoming reluctance	<ul style="list-style-type: none"><li>• Luke</li><li>• R2D2</li><li>• Obi Wan Kenobi</li></ul>	<ul style="list-style-type: none"><li>• R2D2 plays Leia's complete message to Luke</li></ul>	



## 4. Serving XML data

Sharing stories is central to the system. We mean to create an online environment where people can create and share their stories with other humans to communicate and transfer knowledge. Therefore, the data that formalizes the story should be accessible to the parties.

I've looked for Ecore model transformations that could do the job and found [EMF Store](#), [CDO](#), [Teneo](#), [Texo](#) and [EMF-REST](#). The two first options implied in a EMF client side, which implies Java and I wanted more flexibility for my project, maybe making a mobile app in the future. Since the iOS, for example, does not support Java, I have disconsidered both of them. CDO, Teneo and Texo are all related in some way, each with a specific function. Although they seemed more robust than EMF-REST, the latter seemed easier to deal with and the javascript library output seemed like a good thing at the time. Plus the presentation was clearer, and the other three seemed like bloated, confusing projects.

Ultimately, the choice I've made seems like the wrong one. EMF-REST is incomplete, it only supports the GET method. In the end I didn't use the javascript library so it seems like Teneo or Texo would be better fit for the task, although I haven't tried them.

To get the server started you will need

- [Eclipse Java EE IDE for Web Developers](#). (I used Kepler Release)
- Tomcat installed in Eclipse. You may need to configure the Tomcat paths on your project properties under Java Build Path. Under libraries you should add jsp-api.jar and servlet-api.jar.
- EMF-REST (there is a [tutorial](#)<sup>5</sup> to install and test it on Eclipse)

After configuring a project according to the tutorial, download the Ecore model and the sample .xml from the [github project page](#)<sup>6</sup>. Merge the folders to your Eclipse project, right click on the model and select the menu entrance *EMF-REST / Generate All + Copy Libraries*. Run the application by right clicking on your project and selecting the menu entrance *Run As / Run On Server*. The Tomcat will start and the links will be available at port 8080. The output is in JSON format. Find below some test links:

- [localhost:8080/emfrest/app/Database/My1/users/](http://localhost:8080/emfrest/app/Database/My1/users/)
  - A list of users. Notice the user's relationships are not listed.
- [localhost:8080/emfrest/app/Database/My1/users/bernardofbbbraga/](http://localhost:8080/emfrest/app/Database/My1/users/bernardofbbbraga/)
  - A single user from the list
- [localhost:8080/emfrest/app/Database/My1/users/bernardofbbbraga/posts/](http://localhost:8080/emfrest/app/Database/My1/users/bernardofbbbraga/posts/)
  - The posts this user did
- [localhost:8080/emfrest/app/Database/My1/users/bernardofbbbraga/posts/Story/StarWars/elements?depth=3](http://localhost:8080/emfrest/app/Database/My1/users/bernardofbbbraga/posts/Story/StarWars/elements?depth=3)
  - All the elements in the StarWars story. Notice how the `?depth=#` modifies the returned elements to include relationships and their linked elements as well.

---

<sup>5</sup> <http://emf-rest.com/install.html>

<sup>6</sup> <https://github.com/BernardoFBBraga/StoryModelling/tree/master/emf-rest%20components>

## 5. Information visualization

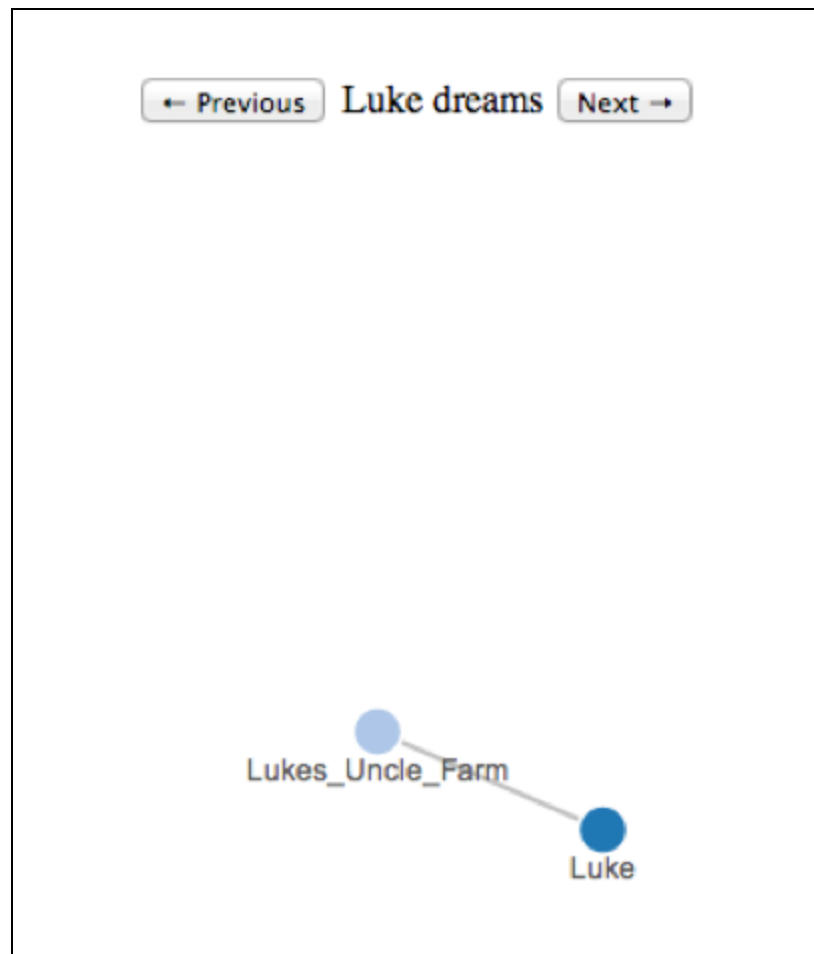
In the previous section we have presented the server side that provides the story data. In this section we present the d3.js script that shows the data on screen. To get the visualization started, download Python and [the web project](#)<sup>7</sup>; on a terminal window navigate to the project folder and execute the following command: `python -m SimpleHTTPServer`

The visualization will be running on `http://localhost:8000/home.html`

The data comes from the EMF-REST server<sup>8</sup>, so be sure to run it first.

To allow the page to access the data, you'll need to configure Tomcat. In Eclipse, under Window>Show View>Servers you will find the Tomcat. Edit the server properties and under monitoring add port 8000 (default used by the Python's simpleHTTPServer).

The script generates a force layout for each situation of the story and you may switch between situations by clicking the Next or Previous buttons. When you do, the script updates the data for the situation and the graph is also updated. The script was based on the d3.js [Force-Directed Graph](#) example and the [Force-based label placement](#).



<sup>7</sup> <https://github.com/BernardoFBBraga/StoryModelling/tree/master/D3js%20graph%20drawing>

<sup>8</sup> warning: links are hard coded, including Tomcat's port number 8080

Figure 2: An example of the running visualization

D3 stands for [Data Driven Documents](#). It is "a JavaScript library for manipulating documents based on data". You may browse through their [gallery](#) to glimpse at the possibilities.

The script works as follows:

First, it downloads the story data and does some processing on it for compatibility. For example, in the StarWars story Luke is present in most situations. We identify it as the same individual in every situation, but in the JSON data, each situation has a copy of Luke's table. So the first processing is removing duplicates of nodes and links, and turning them into references of a single javascript object.

Second, in each link the source and target nodes are replaced for their indexes in a node list. This is the format required by the force layout algorithm.

Two force layouts are created, one will contain all nodes and links of a given situation and the other keeps the node labels and the nodes close together. The array passed as a parameter for the force algorithms is updated as the situations change, removing and adding nodes and labels from the arrays as the story requires.

Please check the next section in conclusions for a discussion on the limitations of the visualization and future directions.

## 6. Conclusions

The system demonstrates the use of EMF Technologies to deploy web applications. In particular, this web application allows the visualization of stories; while their editing is done using the standard Reflective Ecore Model Editor.

There is still much to do to make this prototype ready to test the dissertation's claims but nevertheless it gave much insight in the process. A roadmap for future features include:

- Adding labels to relationships
- Visualization of events
- Visualization of the relation between events and situations
- A timeline for the situation sequence with indication on which situation is selected
- Color coding for different kinds of nodes and links
- The social network around sharing these stories
- Use the OntoUML meta-model for specific reference of models

## Bibliography

1. Marshall McLuhan, *Understanding Media: The Extensions of Man*, McGraw-Hill, New York, NY, 1964.
2. Tuffield, Mischa M., Shadbolt, Nigel R. and Millard, David E. **Narrative as a Form of Knowledge Transfer: Narrative Theory and Semantics**. In, 1st AKT Doctoral Symposium, Milton Keynes, UK, 2005.
3. Campbell, J.: *A Hero with a Thousand Faces*. London: Fontana Press, 1993
4. Sacks, O: **The Man Who Mistook His Wife For A Hat: And Other Clinical Tales**, Simon and Schuster, 1998

## Attachment 1 - Example xmi<sup>9</sup>

```
<?xml version="1.0" encoding="UTF-8"?>
<stories:Database
  xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:stories="http://stories.bernardofbbbraga.com.br"
  xsi:schemaLocation="http://stories.bernardofbbbraga.com.br
../..inputModel/stories.ecore">
  <users username="bernardofbbbraga"
    password="pass124">
    <posts xsi:type="stories:Story"
      creation_time="0006-07-07T00:00:00.000-0300"
      id="StarWars">
      <elements
        xsi:type="stories:Node"
        label="Luke"
        instance_of="Person">
        <performed
          label="Refuses_Obi_Wan's_call"
          happened_in="Reluctance_to_change"/>
        <performed
          label="Luke_travels_to_find_Obi_Wan_Kenobi"
          instance_of="travel"
          happened_in="Increased_awareness"
          motivated_by="plays_Leia's_msg_(cropped)_to_Luke"/>
        </elements>
        <elements
          xsi:type="stories:Link"
          label="at"
          source="Luke"
          target="Lukes_Uncle_Farm"
          instance_of="located_at"/>
        <elements
          xsi:type="stories:Node"
          label="Lukes_Uncle_Farm"
```

---

<sup>9</sup> syntax highlighting powered by <http://tohtml.com/xml/>

```

    instance_of="Local"/>
<elements
  xsi:type="stories:Node"
  label="R2D2">
    <performed
      label="plays_Leia's_msg_(cropped)_to_Luke"
      brings_about="Increased_awareness"/>
    <performed
      label="plays_Leia's_msg_(cropped)_to_Obi_Wan"
      brings_about="Reluctance_to_change"/>
    <performed
      label="plays_full_msg_to_Luke"
      brings_about="Overcoming_reluctance"/>
  </elements>
<elements
  xsi:type="stories:Node"
  label="Obi_Wan_Kenobi"
  instance_of="Person">
    <performed
      label="Asks_Luke_to_follow_him"
      brings_about="Reluctance_to_change"
      motivated_by="plays_Leia's_msg_(cropped)_to_Luke"/>
  </elements>
<elements
  xsi:type="stories:Node"
  label="Obi_Wan_Kenobi's_home"
  instance_of="Local"/>
<elements
  xsi:type="stories:Situation"
  label="Limited_awareness_of_the_problem"
  present="Luke_Lukes_Uncle_Farm at"/>
<elements
  xsi:type="stories:Situation"
  label="Increased_awareness"
  present="Luke R2D2"
  enabled="Luke_travels_to_find_Obi_Wan_Kenobi"
  brought_about_by="plays_Leia's_msg_(cropped)_to_Luke"/>
<elements
  xsi:type="stories:Situation"

```

```
    label="Reluctance_to_change"
    present="Obi_Wan_Kenobi Luke Lukes_Uncle_Farm R2D2"
    enabled="Refuses_Obi_Wan's_call"
    brought_about_by="plays_Leia's_msg_(cropped)_to_Obi_Wan
Asks_Luke_to_follow_him"/>
```

```
  <elements
    xsi:type="stories:Situation"
    label="Overcoming_reluctance"
    present="R2D2 Luke"
    brought_about_by="plays_full_msg_to_Luke"/>
```

```
</posts>
```

```
<posts xsi:type="stories:Model"
  creation_time="0006-07-07T00:00:00.000-0300"
  id="genealogia">
  <classes
    xsi:type="stories:Node_classifier"
    name="Person"
    instances="Luke Mary Obi_Wan_Kenobi John"/>
```

```
</posts>
```

```
<posts xsi:type="stories:Model"
  creation_time="0026-08-07T00:00:00.000-0300"
  id="Traveling">
  <classes
    xsi:type="stories:Node_classifier"
    name="Local"
    instances="Lukes_Uncle_Farm Obi_Wan_Kenobi's_home"/>
```

```
<classes
  xsi:type="stories:Event_classifier"
  name="travel"
  instances="Luke_travels_to_find_Obi_Wan_Kenobi"/>
```

```
<classes
  xsi:type="stories:Link_classifier"
  name="origin"
  source="travel"
  target="Local"/>
```

```
<classes
  xsi:type="stories:Link_classifier"
  name="destination"
  source="travel"
```



```
        target="Local"/>
    <classes
      xsi:type="stories:Link_classifier"
      name="traveler"
      source="travel"
      target="Person"/>
    <classes
      xsi:type="stories:Link_classifier"
      name="located_at"
      instances="at"
      source="Person"
      target="Local"/>
  </posts>
</users>
<users username="jpalmeida"
  password="enschede">
  <posts xsi:type="stories:Story"
    creation_time="0007-07-07T00:00:00.000-0300"
    id="marysstory">
    <elements
      xsi:type="stories:Node"
      label="Mary"
      instance_of="Person"/>
    <elements
      xsi:type="stories:Situation"
      label="sit1"
      present="Mary married_to"/>
    <elements
      xsi:type="stories:Node"
      label="John"
      instance_of="Person"/>
    <elements
      xsi:type="stories:Link"
      label="married_to"
      target="Mary"/>
  </posts>
</users>
</stories:Database>
```