An Ontology–Based Approach to Text–to–Picture Synthesis Systems

Dmitry Ustalov, Aleksandr Kudryavtsev

Ural Federal University, Yekaterinburg, Russia
dmitry@eveel.ru, vt@dpt.ustu.ru

Abstract. In this paper, we present an ontology–based approach to text–to–picture synthesis. Our approach operates with an ontology in the RDF/XML format. This provides loose coupling of the system components, unification of the interacting objects representation and their behaviour, and makes possible verification of system information resources.

Keywords: text–to–picture, text–to–scene, natural language processing, depiction rules, semantic representation, Web Ontology Language, Resource Description Framework.

1 Introduction

A picture is worth of a thousand words. The text–to–picture synthesis problem is actual because of existence of many domains where clearness of textual information is necessary: foreign language learning [12], traffic accident visualization [1], rehabilitation of people with cerebral injuries [4], etc.

Utkus [10] is a text–to–picture synthesis system (TTP system) that is developed since 2011 and is designed to work with small texts of 1–3 Russian sentences: fragments from children literature, microblog posts, news summaries, comments on Web–sites. These texts are suitable for automatic processing and further visualization [13]. The current work differs from previous TTP systems in its focus on conveying the gist of general, semantically unrestricted Russian language text.

TTP systems have three stages of processing [2]:

1. A stage of *linguistic analysis* — tokenization, morphological and syntactic parsing, obtaining the semantic representation of the input text;
2. A stage of *depictors generation* — generation of the set of graphical depictors that corresponds with obtained semantic representation;
3. A stage of *picture synthesis* — construction of vector or raster image from the graphical primitives that are positioned following the generated depictors.

In TTP systems, every processing stage strongly depends on many information resources, including:

- Thesaurus that containts words and their relations (synonymy, hyponymy, etc);

- Gallery that contains different graphical primitives for interacted objects (actors), which becomes rendered in the final images;
- Depiction rules that define how one or many actors can be depicted into the final images;
- Frames, which describe allowed properties of actors.

The volume and complexity of these resources are high. Therefore, TTP systems must have a straight way to connect such resources during the text processing.

2 Related works

There are several full–functional analogues that are described in various papers [1, 2, 4, 5, 11, 13]. Unfortunately, an approach to unification the information resources is presented only in [2]. That paper presented the WordsEye system, which builds 3D scenes by with certain descriptive English sentences, e.g., “The huge head is on the tan horse. The horse is on the extremely tall mountain range. The fence is 10 feet behind the horse. The fence is 50 feet long.”

The following desicions are made in the WordsEye system:

1. WordNet thesaurus is used to identify the semantic relations between separate words;
2. During the text processing, specially defined frames are mapped into the found syntactic groups to obtain additional information about actors: colour, size, etc;
3. Behaviour that is implemented in known actions (verbs), and is described in *depiction rules*, which are defined in a declaration–style Lisp program;
4. A proprietary Izware Mirai 3D animation system is used with Viewpoint Model Library to perform visualization problem.

Note two significant drawbacks of these decisions:

1. Despite of rich possibilities of the Lisp programming language, usage of this language complicates the replenishment of depiction rules set because of high requirements of developers experience;
2. Work in 3D demands considerable efforts and resources, which are not justified by final quality: in most cases, it is enough to deal with 2D images [12].

3 Suggested approach

Similarly to [2], we consider actors in terms of object paradigm:

1. Actors have properties: colour, etc;
2. Actors have methods: functions that reflect actors relations: *to fall*, *to lay*, etc.

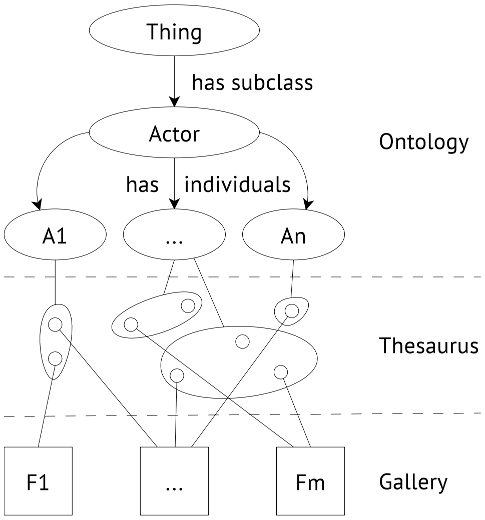


Fig. 1. Connection of ontology, thesaurus, and gallery

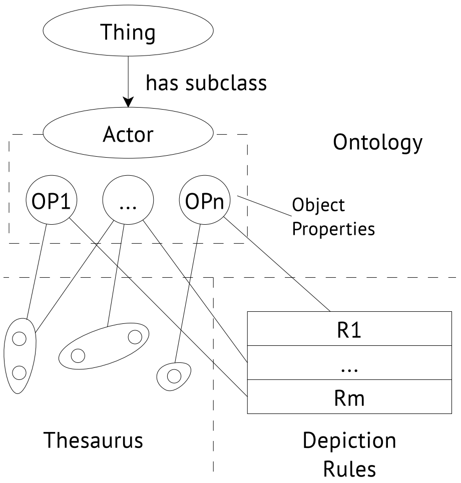


Fig. 2. Connection of ontology, thesaurus, and depiction rules

We propose to formalize into an ontology all the knowledge about actors: their possible characteristics and relations. We also propose to split the ontology, thesaurus, depiction rules and gallery to provide loose coupling of these components of the TTP system (Fig. 1, 2):

- Words and their semantic relations are represented in a thesaurus;

- Several figures from gallery can be associated with each word in thesaurus: despite words *tomcat* and *cat* are antonymes by gender, they both are hyponymes to word *animal*;
- Ontology has the class *Actor*, and instances of this class are linked to synsets in thesaurus. Therefore, for every *set of synsets* an *Actor* instance can be defined by correspondent properties;
- There are defined *object properties* for *Actor* instances. These object properties are associated with verb synsets in thesaurus and represent all possible relations among actors (i.e., `fall(actor)` and `fallTo(actor1 actor2)`);
- Also, there *data properties* are defined and represent different parameters of actors (e.g., colour);
- Depiction rules that specify the behaviour of each object property (Fig. 2) are defined in a separate XML document.

Elements of ontology are linked to thesaurus synsets using the OWL annotation mechanism. It is important to note that one element can be linked to many synsets. These synsets can belong to thesauri of different language because of internationalization method that is implemented in OWL.

3.1 Examples

The *Actor* class is a direct subclass of the *Thing* class:

```
<owl:Class
  rdf:about="http://utkus.eveel.ru/World.owl#Actor"/>
```

Instances of the *Actor* class (Fig. 3) can be linked to synsets using OWL annotations:

```
<owl:NamedIndividual
  rdf:about="http://utkus.eveel.ru/World.owl#Man">
    <rdf:type
      rdf:resource="World.owl#Actor"/>
    <synset xml:lang="ru">2039</synset>
    <synset xml:lang="ru">2040</synset>
    <synset xml:lang="ru">238</synset>
    <synset xml:lang="ru">6939</synset>
    <synset xml:lang="ru">75</synset>
</owl:NamedIndividual>
```

Object properties are also defined for the *Actor* class, and they represent all the predicates that are operated by the system. *Object properties* are connected with depiction rules that specify their behaviour.

Equivalence relation is possible between *object properties*. In our approach, the SPO–triples[1] `(fall man)` and `(fall man chair)` will be attributed to different *object properties*: `fall` and `fallTo`.

[1] SPO — a tuple of predicate, subject, and object.

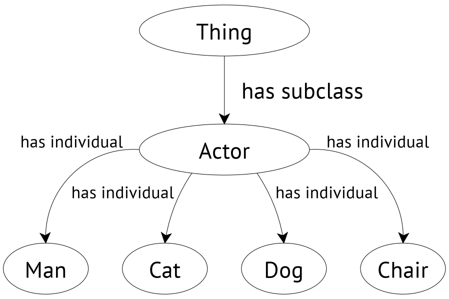


Fig. 3. Ontology fragment with *Actor* instances

```
<owl:ObjectProperty
  rdf:about="http://utkus.eveel.ru/World.owl#fall">
    <synset xml:lang="ru">106</synset>
    <synset xml:lang="ru">397</synset>
    <synset xml:lang="ru">406</synset>
    <rdfs:domain
      rdf:resource="World.owl#Actor"/>
    <owl:equivalentProperty
      rdf:resource="World.owl#fallTo"/>
</owl:ObjectProperty>

<owl:ObjectProperty
  rdf:about="http://utkus.eveel.ru/World.owl#fallTo">
    <synset xml:lang="ru">106</synset>
    <synset xml:lang="ru">397</synset>
    <synset xml:lang="ru">406</synset>
    <rdfs:domain
      rdf:resource="World.owl#Actor"/>
    <rdfs:range
      rdf:resource="World.owl#Actor"/>
</owl:ObjectProperty>
```

To represent detected *object properties* on the final picture, it is necessary to assign the specific behaviour to each known *object property*. This behaviour is specified by *depiction rules* which are declared in a separate XML document. For the `fallTo` *object property* we have the following *depiction rule*:

```
  <rule rdf:about="http://utkus.eveel.ru/World.owl#fallTo">
    <rotate>
      <yield id="subject" />
      <yield id="object" />
    </rotate>
    <together>
    <yield id="subject" />
    <yield id="object" />
  </together>
</rule>
```

In this example, the *subject* and *object* of the predicate would be put together, and the *subject* will be diverted onto *object*.

4 Implementation

The Utkus prototype under discussions was written on the Ruby programming language:

1. Link Grammar for Russian syntactic parser [6] is used because of its availibility and easy parseable format;
2. Only verb phrases and related noun phrases are extracted from the dependency tree of each sentence of the source text. These syntactic groups are mapped into the SPO–triplets;
3. Ontology is defined in the RDF/XML format using the Protegé editor;
4. There are only synsets in our Russian dictionary [9]: no hyponyms, etc;
5. Gallery is composed by sprites from The Noun Project [7] collection. These sprites are cropped, rasterized, and associated with noun synsets;
6. Final rendering is performed using GD2 library in form of PNG raster images of 640 × 480.

As example, there are four images that been generated by Utkus system. With a view of place economy, these images been cropped. These images (Fig. 4(a), 4(b), 4(c), 4(d)) are correspond to texts:

1. A man has fallen into the fire[2];
2. Several houses[3];
3. There are a man and a woman in the house[4];
4. A certificate, a bear, a rain[5].

It should be noted that Utkus system is unable to represent plural words (Fig. 4(b)) at this moment.

5 Conclusion

We have presented the approach to organize the TTP system information resources. This approach provides loose coupling of ontology, thesaurus, gallery and depiction rules.

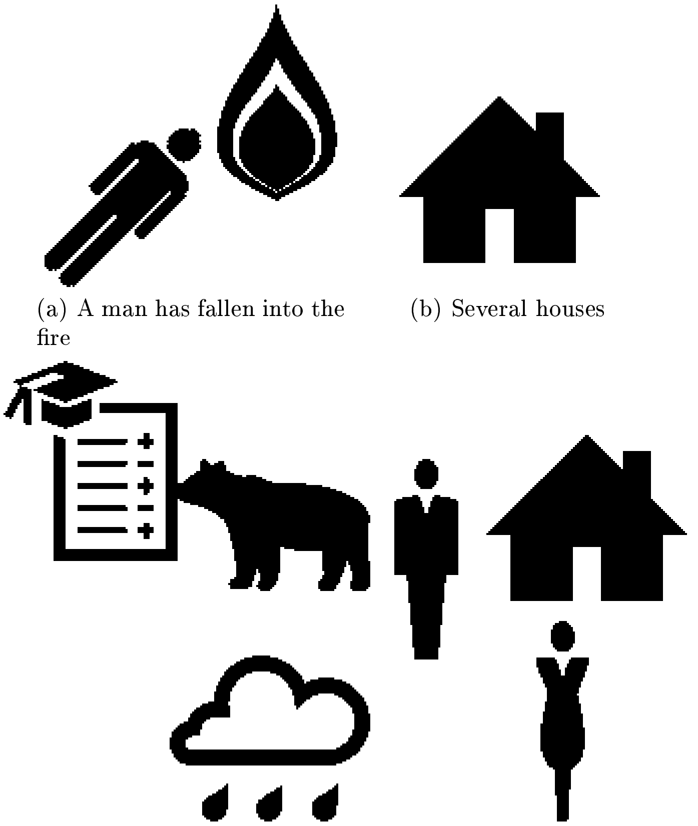
Main advantages of this approach are:

[2] Человек упал в огонь, in Russian.

[3] Несколько домов, in Russian.

[4] В доме находились мужчина и женщина, in Russian.

[5] Аттестат, медведь, дождь, in Russian.



(a) A man has fallen into the fire

(b) Several houses

(c) A certificate, a bear, a rain

(d) There are a man and a woman in the house

Fig. 4. Depiction of the texts

1. Simplicity of development and modification all the information resources that are used by TTP system:
   - Ontology can be modified with any available ontology editor (e.g., Protegé);
   - Depiction rules can be edited with any text editor, or any XML editor;
   - Thesaurus and gallery data can be modified as any data in relational database (in our implementation, PostgreSQL is used).
2. RDF/XML ontology allows one to reuse these resources in other applications and domains;
3. Verification instruments (such as inference systems) can help us to control the quality of information resources.

Figures 4(a), 4(b), 4(c) are produced during testing our Utkus TTP system under development. The Utkus TTP system is based on this approach.

5.1 Future Work

We have several reasons for future work:

1. To switch to the full–featured thesaurus to unify the thesauri resources (e.g., Russian WordNet [8]);
2. To enhance the linguistic analysis subsystem to handle such parts of speech as adjectives, pronouns, numerals, etc;
3. To solve the problem of predicate ambiguation [3] when generating the semantic representation;
4. To perform experiments on the Utkus prototype and make changes in the system components, if necessary.

Acknowledgements. Authors would like to thank the Institute of Mathematics and Mechanics UrB RAS for the provided computer equipment.

References

1. Åkerberg, O., Svensson, H., Schulz, B., Nugues, P.: CarSim: an automatic 3D text-to-scene conversion system applied to road accident reports. In: Proceedings of the 10th Conference on European Chapter of the Association for Computational Linguistics–Volume 2. pp. 191–194. Association for Computational Linguistics (2003)
2. Coyne, B., Sproat, R.: Wordseye: an automatic text-to-scene conversion system. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. pp. 487–496. ACM (2001)
3. Fomichov, V.: A comprehensive mathematical framework for bridging a gap between two approaches to creating a meaning-understanding web. International Journal of Intelligent Computing and Cybernetics 1(1), 143–163 (2008)
4. Goldberg, A., Rosin, J., Zhu, X., Dyer, C.: Toward text-to-picture synthesis. In: NIPS 2009 Mini-Symposia on Assistive Machine Learning for People with Disabilities (2009)
5. Li, H., Tang, J., Li, G., Chua, T.: Word2image: Towards visual interpretation of words. In: The 16th ACM International Conference on Multimedia (2008)
6. Link Grammar for Russian, `http://slashzone.ru/parser/`
7. NounProject, `http://thenounproject.com`
8. Russian Wordnet, `http://www.wordnet.ru`
9. Russian Language Dictionaries, `http://speakrus.ru/dict/index.htm`
10. Utkus, `http://utkus.eveel.ru`
11. Yamada, A., Yamamoto, T., Ikeda, H., Nishida, T., Doshita, S.: Reconstructing spatial image from natural language texts. In: Proceedings of the 14th Conference on Computational Linguistics–Volume 4. pp. 1279–1283. Association for Computational Linguistics (1992)
12. Yoshii, M., Flaitz, J.: Second language incidental vocabulary retention: The effect of text and picture annotation types. CALICO journal 20(1), 33–58 (2002)
13. Zhu, X., Goldberg, A., Eldawy, M., Dyer, C., Strock, B.: A text-to-picture synthesis system for augmenting communication. In: Proceedings of the National Conference on Artificial Intelligence. vol. 22, p. 1590. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999 (2007)