

My Dramatis Fabrica : an Ontology for Annotating and Performing Playscripts

Remi Ronfard¹ and Federico Uliana²

1 Inria, LJK, University of Grenoble, France
remi.ronfard@inria.fr

2 Inria, LIRMM, University of Montpellier, France
federico.uliana@inria.fr

Abstract

We describe an ontology for annotationg, visualizing and performing play scripts.

Thoe goal

1998 ACM Subject Classification Dummy classification – please refer to <http://www.acm.org/about/class/ccs98-html>

Keywords and phrases Dummy keywords – please provide 1–5 keywords

Digital Object Identifier 10.4230/OASISs.xxx.yyy.p

1 Introduction

A playscript is a written version of a play or other dramatic composition; used in preparing for a performance.

In this paper, we describe an ontology suitable for annotating playscripts to make them machine-readable. This is meant as an initial step for generating virtual dramatic performances by computer.

As a preliminary step, we demonstrate that MDF makes it possible to describe an existing performance in the words of the play script.

We briefly present all formalisms at the basis of MyDF.

The prose storyboard language (PSL) is a formal language for describing movies shot by shot, where each shot is described with a single sentence, like “*Dolly on Brandon after he crosses the door*”. The language describes the spatial structure of individual movie frames and their temporal sequence.

The Movie Script Markup Language (MSML) is a language for describing the structural representation of screen play narrative. A scrip is decomposed in MSML in a set of dialogues and action blocks. For example “*Joe grabs the candle*” or “*Andi says -I’ll make it-*”.

These declarative languages are seen as complementary tools for describing narrative stories. MSML describes the structure of a movie script, while PSL says how to turn it into shots. Both languages attempt to formalize how a script can become a movie, by describing actions, entities, and places drawn from a storyworld.

In 3D animation each storyworld entity is interpreted by a 3D object.

We call *Automatic Casting (AC)* the problem of providing a set of 3D objects that can be used to interpret storyworld entities described by a script. The key to address this problem is to extend PSL and MSML with explicit connections between storyworld entities and 3D models.



© Ronfard and Uliana;
licensed under Creative Commons License CC-BY
Conference/workshop/symposium title on which this volume is based on.
Editors: Billy Editor and Bill Editors; pp. 1–8



OpenAccess Series in Informatics
OASIS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A natural approach for doing automatic casting is to associate each storyworld entity with a 3D object, which is retrieved at the moment of the construction of the scene. This is however of limited interest because *i*) it has to be done manually and *ii*) it lacks the flexibility of selecting *candidate* 3D objects for characters and props based on different criteria.

In this work, we present a novel ontology-based approach for automatic casting based on an ontology of storyworld entities and 3D objects.

By introducing functional descriptions of 3D objects, casting-specific inference rules, we obtain a declarative framework that allows to infer which 3D objects are skilled enough (like, real actors!) to play the movie roles.

1.1 Technical issues

Does a place contain objects ? Or is it sufficient to state that objects are in a place ?

Relations :

Action continues action : previous action by the same agent.

Action follows action : previous action in the play script.

Action resultof action : action which causes this new action

Actor *plays_role* Character

Prop *stands_for* Object

StageAction mimics StoryAction

StageLocation *stands_for* Place

Performance mimics Story

What is the right way to introduce dramatic entities ?

Story entity -> *played_by* -> StageEntity

Attributs d'un acteur -> liste de skills ? mesh 3D ?

Actor -> MovieActor, TheatreActor, VirtualActor

Prop -> MovieProp, TheatreProp, VirtualProp

StageLocation -> MovieStageLocation, TheatreStageLocation, VirtualStage

StageAction -> MovieAction, TheatreAction, VirtualAction = Animation

we can add

ScreenAction -> as recorded by a real or virtual camera

ScreenLocation -> as recorded by a real or virtual camera

2 Related work

MSML describes the structure of a movie script or theatre play [10]

PSL describes movies shot by shot [8]

Damiano et al introduce the "drama ontology" [5]

Eno et al. stress the importance of providing ontology services in virtual reality [6].

NarrativeML was proposed by as a tool for annotating narrative texts [7].

OntoFilm [3] is a core ontology for film production which provides a standardized model which conceptualizes the domain and workflows used at various stages of the film production process starting from pre-production and planning, shooting on set, right through to editing and post-production.

Cataldi et al. introduced an ontology for representing the dramatic features of narrative media (video, text, audio, etc.), focused on the notions of the character's motivated actions [2].

Carriave used an ontology of film events to generically describe elements of television collections and segment their instances (e.g. tv news broadcasts or television series episodes) [1].

SUMO is an upper level ontology

The movie ontology MO ¹ aims to provide a controlled vocabulary to semantically describe movie related concepts (e.g., Movie, Genre, Director, Actor) and the corresponding individuals (?Ice Age?, ?Drama?, ?Steven Spielberg? or ?Johnny Depp?. The Web Ontology Language (OWL) is used to specify the MO ontology. Several other ontologies that are provided in the Linked Data cloud are considered and integrated to highly couple the MO ontology with the Linked Data cloud to take advantage of synergy effects.

The Cinema Ontology Project ² is a knowledge organization construction project. The goal of the project is to produce an ontology¹, which is a formalized specification, of film domain discourse and knowledge. The film domain includes: individual films and cinema collectively, filmmaking, filmmakers, film industry, and film culture, such as film viewing, film criticism, film study, and film "fandom", etc.

Cavell's reflection on the ontology of film

Julie Books : The ontology of film

There has been limited work on the structure of stage-plays and screen-plays.

A founding document in the contemporary study of the screenplay is Claudia Sternberg's "Written for the Screen". Other books include Steven Maras's "Screenwriting", Steven Price's "The Screenplay", J. J. Murphy's "Me and You and Memento and Fargo" and Jill Nelmes's anthology "Analysing the Screenplay".

Owl-Blender ³ takes a properly annotated OWL file as input and outputs a Blender file to create a 3D visualization of the ontology.

The theatre ontology ⁴ is an ontology for organizing theatrical data.

Ujilings describes a system for generating stories in a virtual environments based on an ontology of events and objects [9].

Cua et al [4] use SUMO to represent storytelling knowledge and query actions and events that may take place in a story.

3 MyDF Ontology

Storyworld entities and 3D objects in the MyDF ontology are now described.

In the spirit of the Semantic Web, our storyworld ontology is built on the SUMO ontology, in the sense that it *links* and *reuses* part of its vocabulary, for entities, places, and actions. To achieve this, we use as ontological language RDF(S), the standard for Semantic Web data.

An ontology is a formal description of a domain of interest, written by means of classes, properties, and instances. The MyDF ontology contains top-level classes like *mdf:Actor* and *mdf:Action*. The namespace *mdf:* allows to distinguish MyDF resources in the Linked Data.

The RDF(S) language allows to write ontologies by means of triples of the form (*s p o*) where *s*, *p* and *o*, denote the *subject*, *property* and *object* of the triple, respectively. An example of RDF(S) triple making the correspondence between MyDF and SUMO concepts is

(*mdf:Actor owl:sameAs sumo:actor*)

¹ <http://www.movieontology.org/>

² <http://jedfilm.com/cinema-ontology>

³ <http://sourceforge.net/projects/owl-blender/>

⁴ <http://lukeblaney.co.uk/semweb/theatre>

We stress also the fact that building on SUMO also allows our framework to easily deal with synonyms and be multilanguage so as to handle a larger class of movie scripts.

4 Storyworld entities

The MyDF ontology allows to encapsulate the storyworld entities, namely characters, props, places, and actions, found in a PSL or MSML script. For instance, the PSL instruction “*Dolly on Brandon after he crosses the door*”, indicates three entities: Brandon, a door, and the action of crossing the door. Similarly, the MSML annotation “*Joe grabs the candle*”, indicates Joe, a candle, and the action of grabbing the candle.

How are these classified in the MyDF ontology?

4.1 Story characters

Characters like Brandon or Joe are seen as instances of the class *mdf:Character*. This is denoted by the triple (*mdf:Brandon* *rdf:type* *mdf:Character*). The *rdf:type* property is used to denote the membership of an element to a class.

4.2 Story objects

Differently from actors, the concepts for the props like the candle or the door, are classes directly drawn from the SUMO ontology. In this case, *sumo:candle* and *sumo:door* we also drawn from the SUMO taxonomy the fact that both are subclasses of *sumo:device*, *sumo:artifact*, and *sumo:object*. This knowledge is extremely useful for automatic casting. For example, if the scripts says that “*Bob grabs an object and hits his enemy*” the system can suggest a candle as the object needed to play the scene. All props drawn from SUMO are organized in a taxonomy whose root concept is *mdf:Props*.

4.3 Story places

Places are also drawn from SUMO. For example, the PSL instruction “*MCU on Alice as she enters the room*” designates a place, which is the *room*. The corresponding concept in our ontology will be *sumo:room*. All places drawn from SUMO are organized in a taxonomy whose root concept is *mdf:Place*.

4.4 Story actions

Actions in a script are built around a verb, like “*Joe grabs the candle*”. As before, we draw from SUMO the concepts that we use to describe actions. For instance, we import into MyDF the class *sumo:grabbing* but also its superclass *sumo:touching*. As for props, this information is crucial for realizing the automatic casting. All concepts of actions are again organized in a taxonomy where the root concept is *mdf:Action*.

4.5 Story events

Once characters, props, places, and actions are in place, we can finally describe complex script events. All events are seen as instances of the *mdf:Event* class. In RDF(S) the event “*Joe grabs the candle in the room*” is translated with the following set of triples.

```
( mdf:act1 rdf:type mdf:Action )
( mdf:act1 mdf:doneBy mdf:Joe )
( mdf:Joe rdf:type mdf:Character )
( mdf:act1 mdf:actionPerformed sumo:grabbing )
( mdf:act1 mdf:receivedBy sumo:candle )
( mdf:act1 mdf:where sumo:room )
```

Each script event is referred by a unique RDF resource, in the example *mdf:act1*. We then employ several domain-specific properties to denote who performs the action (*mdf:doneBy*), which action is performed (*mdf:actionPerformed*), who or what complements the action (*mdf:receivedBy*), and the place where the action takes place (*mdf:where*).

This can also be seen as a graph.

It is worth noting that RDF(S) makes easy for our ontology to be compatible with SUMO, because it allows to refer to classes, properties, and instances with no distinction. With a description logic approach, it would not be possible for instance to relate an instance with a class with a domain-specific property like (*mdf:act1 mdf:actionPerformed sumo:grabbing*), unless looking at formalisms of the higher-order.

5 Stage entities

The MyDF ontology also describes actors and props on a virtual stage with functional description that allow to infer whether they are skilled enough to play a role in the script.

To illustrate, consider the script indication “*Joe kneels down*”. When doing automatic casting, one should search for a 3D object representing a man that is complex enough so as to perform the action of knelling. This may require for instance that the 3D object has a deformable skeleton. However, for mimicking simple actions like “*Bob falls*” the 3D object can be arbitrarily simple.

Objects are represented in MyDF with sets of triple, in a similar way to actions.

```
( mdf:obj1 mdf:propertyObject3D mdf:canRotate )
( mdf:obj1 mdf:propertyObject3D mdf:hasSkeleton )
( mdf:obj1 mdf:describes sumo:man )
```

5.1 Stage actors

Stage actors can be real actors or virtual actors

Virtual actors are 3D animated characters that can be used to play a character’s part.

Real actors can be theatre actors playing on a theatre stage, or movie actors playing on a movie set.

5.2 Stage props

Virtual props are 3D objects that can be placed on a virtual stage.

5.3 Stage locations

Real stage location can be locations on a theatre stage, or locations on a movie set.

Virtual stage locations are locations on a virtual stage.

5.4 Stage actions

Stage actions are performed by actors depending on their acting skills

Virtual actor performances are 3D animations.

5.5 Stage events

Stage events are happenings, including stage actions

6 Performing play scripts

The general problem of directing a performance is to find correspondences from story events to stage events.

6.1 Decorating the stage

Find correspondences from story places and objects to stage sets and props.

6.2 Casting actors

Once 3D objects are specified, the idea is to define their skills as a consequence of their functional description. This is equivalent to writing *casting-rules* for actors and objects.

For instance, we can say that any 3D object x that has a skeleton can do a downward-motion. This translates into the following inference rule

$$(x \text{ mdf:propertyObject3D mdf:hasSkeleton}) \longrightarrow (x \text{ mdf:canMimickAction sumo:downwardMotion})$$

which is interpreted as follows. The rule makes infer a new RDF triple for each 3D object described in the ontology that has a skeleton. This further allows to query the ontology and retrieve all 3D models that can do a *downward-motion*. Moreover, by evaluating the rule

$$(x \text{ mdf:canMimickAction } y), (z \text{ rdfs:subClassOf } y) \longrightarrow (x \text{ mdf:canMimickAction } z)$$

we can infer for instance that any model that can do a downward-motion can do also a more specific mouvement declared as one of its subclasses. By exploiting the SUMO taxonomy, one can infer that any model that has a skeleton can also *kneel-down*, or *fall* or *sitting-down*.

Notice that this is a declarative approach allow to change any time the *casting rules*, and thus the skills that an object has to meet in order to be a candidate for a role. For instance, having a rich collection of 3D object can permit to have more refined rules, while for a small collection of objects one can aim at putting in place a scene with a minimum number of distinct objects.

The inference rules we presented are called Datalog rules.

6.3 Directing actors

When all story objects, characters and places have been assigned to stage props, actors and settings, it becomes possible to translate the story into a dramatic score with all actor actions synchronized together.

Following established practices in the theatre, we define a set of "cues", which act as temporal signals for synchronizing actors and events together.

One benefit of our representation is that the director can choose to name "story world entities" or "stage entities", and the system can translate between the two representations. If the names are chosen appropriately, the director can even mix story and stage entities at will.

More importantly, it is quite straight-forward to reorganize the stage with different props and settings, and to re-cast actors, and automatically update the dramatic score accordingly.

7 Filming performances

When a performance is recorded by a real or virtual camera and projected on a screen, it results in screen actions and events.

7.1 Screen events

7.2 Screen actions

7.3 Screen locations

8 Experimental results

We present results on several possible stagings of a short scene from the screenplay (play text) of the movie "back to the future", written by Robert Zemekis.

We show examples of stage events reconstructed in 3D animation with virtual actors, props and stages.

We show examples of screen events in an animated movie generated from this 3D animation.

We show examples of screen events in the real movie.

Future work will investigate the generation of performances of other play scripts borrowed from theatre, motion pictures and 3D animation.

Maybe the opening scene of death of a salesman ?

Maybe the opening scene of cat on a hot tin roof ?

Maybe the patio scene from notorious ?

Maybe the opening scene of psycho ?

Maybe a scene from endless night ?

Maybe a scene from the artist ?

Maybe a scene from "la ronde" ?

9 Limitations and future work

10 Conclusion

References

- 1 Jean Carrive, Francois Pachet, and Remi Ronfard. Using Description Logics for Indexing Audiovisual Documents. In *International Workshop on Description Logics (DL '98)*, Trento, Italy, June 1998.
- 2 Mario Cataldi, Rossana Damiano, Vincenzo Lombardo, and Antonio Pizzo. Representing dramatic features of stories through an ontological model. In *Proceedings of the 4th International Conference on Interactive Digital Storytelling, ICIDS'11*, pages 122–127, Berlin, Heidelberg, 2011. Springer-Verlag.

- 3 Ajay Chakravarthy, Richard Beales, Nikos Matskanis, and Xiaoyu Yang. Ontofilm: A core ontology for film production. In *Proceedings of the 4th International Conference on Semantic and Digital Media Technologies: Semantic Multimedia*, SAMT '09, pages 177–181, Berlin, Heidelberg, 2009. Springer-Verlag.
- 4 Jeffrey Cua, Ethel Ong, Ruli Manurung, and Adam Pease. Representing story plans in sumo. In *Proceedings of the NAACL HLT 2010 Second Workshop on Computational Approaches to Linguistic Creativity*, CALC '10, pages 40–48, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
- 5 Rossana Damiano, Vincenzo Lombardo, and Antonio Pizzo. Formal encoding of drama ontology. In *Proceedings of the Third International Conference on Virtual Storytelling: Using Virtual Reality Technologies for Storytelling*, ICVS'05, pages 95–104, Berlin, Heidelberg, 2005. Springer-Verlag.
- 6 J.D. Eno and C.W. Thompson. Virtual and real-world ontology services. *Internet Computing, IEEE*, 15(5):46–52, Sept 2011.
- 7 Inderjeet Mani. *Computational Modeling of Narrative*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2013.
- 8 Rémi Ronfard, Vineet Gandhi, and Laurent Boiron. The prose storyboard language: A tool for annotating and directing movies. In *Workshop on Intelligent Cinematography and Editing, Foundations of Digital Games, May, 2013*, Chania, Crete, Grèce, May 2013. Society for the Advancement of the Science of Digital Games.
- 9 J.R.R. Uijlings. Designing a virtual environment for story generation. Master's thesis, University of Amsterdam, 2006.
- 10 Dieter Van Rijsselbergen, Barbara Van De Keer, Maarten Verwaest, Erik Mannens, and Rik Van de Walle. Movie script markup language. In *Proceedings of the 9th ACM Symposium on Document Engineering*, DocEng '09, pages 161–170, New York, NY, USA, 2009. ACM.