Visualization of character's intentions in dramatic media

Vincenzo Lombardo, Rossana Damiano and Antonio Lieto
CIRMA and Dipartimento di Informatica
Università di Torino
Torino, Italy
Email: {vincenzo, rossana, lieto}@di.unito.it

Antonio Pizzo
CIRMA and Dipartimento di Studi Umanistici
Università di Torino
Torino, Italy
Email: antonio.pizzo@unito.it

***Abstract*—The representation of characters' intentions in a story is of great importance for media scholars and analysts, and it is susceptible of applicative scenarios within the media industry. In this paper, we introduce an interactive system for the visualization of a story analysis based on a plan-based representation of the characters' intentions. The system relies on an ontology of drama and builds upon the unrestricted annotation provided by narrative enthusiasts and media students. The system is able to build the mapping between a library of plans and the users' annotation, and to visualize the contributions of the several characters' intentions to the whole plot. The system was tested on the analysis of a short movie.**

I. Introduction

There exist a variety of areas of creativity that concern events unfolding over time and occurring in some space. All these areas usually benefit from a narrative mapping, i.e. a visualization of the events through a combination of "information graphics, journalistic diagramming, visualizations, reconstructions, and some conventional-looking (but ambitious) geographic maps" [1]. The aim is towards a representation of the event structure, with the possibility of carrying out analyses for specific aspects of the narrative. The result is the creation of an information space where the bits and pieces of some endeavor find an appropriate place in some structure and the design of novel interfaces for the exploration of such a space. For example, Narratives [2] is a system for viewing temporally-changing data based on keyword visualization, working with a corpus of blog entries that talk about news stories. The visualization relies upon a line graph, with users that can interact to see what additional concepts are most associated with a selected term. In other approaches, the aim is the generation of stories. Narrative theatre [3] is a tool for supporting the creation of fables. It relies on a computational framework that leverages the knowledge about the writing domain in order to reason about the events and create a visual representation of each event. It mostly focuses on the creation of storyboards from the written text.

This paper presents a tool for improving the didactics about the dramatic media through a visualization of the content of a media object. In particular, we focus on the visualization of the analysis of the characters' intentions (represented by plans) as a result of the interpretation of the actions in an audiovisual fragment. Dramatic media [4] typically involve a narration based on the live action of characters in conflict. Linear audiovisuals involve a timeline along which the dramatic incidents unfold, supported by the characters' motivations. These motivations, called intentions and pursued through plans, are usually arranged on a tree, with component plans or action as children of some node representing a wider and longer standing plan. Therefore, the challenges posed by our visualization problem concern the display of a timeline, with a fixed order of the component of incidents, and the superimposition of trees that represent the characters' intentions. However, incidents and plans should be aligned to reveal the structure of motivations at the base of the incident unfolding in the plot.

Tree layout, especially in the case of multiple trees spanning the same set of basic elements (usually the leaves of a tree) has been the object of several approaches of information visualization (see the survey in [5] on single and multiple trees). Node-link, nested squares or circles, horizontal and vertical adjacency, indented–list, and matrix representations are well known in the literature, each with specific advantages and disadvantages, depending on the task at hand. For example, containment (or nested) approaches have the advantage of a bounded space but leave no room for node content visualization. Some work [6] has also addressed the problem of stitching together hierarchical structure and time into one visualization space, in order to help an analyst understand how very large hierarchies change through time; the goal is to enable the analyst to detect patterns of relationships. In our case, the interest is in the visualization of multiple trees that span the same frontier. In particular, here we refer to the timeline of incidents that occur in a narrative plot (leaf nodes of a tree), that result from the projection of the characters' plans (internal nodes). Since, the several characters' intentions are hierarchically organized into overlapping trees the necessity of multiple tree visualization arises.

The structure of the paper is the following. In the next section we sketch the computational ontology Drammar that represents the facts about drama (what concepts are to be included, what are the relations among the concepts, how we can build an ontology for a specific drama, i.e. the annotation of a dramatic media object). Then, we also introduce in detail

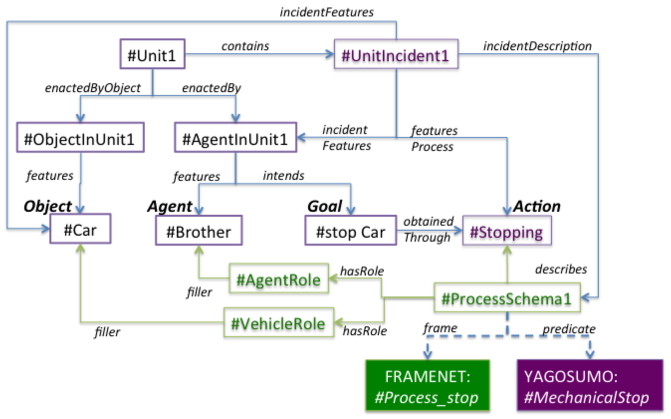

Fig. 1. The annotation of the incident where Brother stops Car, with instances of the ontological classes Unit, Entity (Agent and Object), Goal, Incident, Action (Process), and references to the external ontology YagoSumo and the linguistic resource Framenet.

the specific modeling issues related to actions and plans, and the timeline concepts of the elements that actually appear in the audiovisual fragment, and how they are mapped one on the other in order to build a useful visualization. The core of the paper consists of the visualization tool, that provides an interactive exploration of the characters' intentions and their mapping on the timeline incidents. Finally, we provide a preliminary, qualitative evaluation of the tool in the didactics of dramatic media analysis, and discuss the results. Conclusions end the paper.

II. Drama ontology and annotation

In this section we introduce the Drammar ontology and the annotation schema employed in project CADMOS through an example (see [7] for details). The ontology describes the content and structure of a story in terms of Units (the segments of a story), Entities (i.e., Agents and Objects involved in the story actions), and Relations (action structures relating the entities one another), generalizing over the specific format by which it is expressed (short movie, novel, screenplay, etc.) and the medium through which it is conveyed. The ontology refers to large–scale semantic resources for the description of the commonsense knowledge: the two upper ontologies Suggested Upper Merged Ontology (SUMO, [8]) and Yet Another Great Ontology (YAGO [9]), merged into YAGO–SUMO [10], which provide very detailed information about millions of situations, including entities (agents and objects), processes/actions, and events, and FrameNet [11], describing situations, processes/actions, and/or events through a semantic template that depicts the situation in terms of roles played by the elements which participate in it.

We describe the annotation of a story incident (see Figure 1), driven by the Time Indexed Situation design pattern developed in the descriptive ontology DOLCE [12]. This incident is extracted from the short movie "Exit strategy", originally produced for project CADMOS, to evaluate the consequences of a semantic annotation onto the production methods. This "noir" story concerns a group of three people, Brother, Sister, and (Sister's) Girlfriend, who are going (by car) to spend a weekend in the country villa of Brother and Sister's parents. The car trip draws attention to the arrogant attitude of Brother, who also plays a bad joke to an unlucky Man, asking for help because of his broken truck. Figure 1 shows the representation of the incident in which Brother stops the car, to play a joke to the Man (see below). The unit, called *#Unit1*, features two entities (via the *#incidentFeatures* property), the agent *#Brother* and the object *#Car*. The Unit contains a UnitIncident *#UnitIncident1*, which features (via the *featuresProcess* property) a stopping action/process *#Stopping* and an Agent (*#Brother*, through the *AgentInUnit* class), who *intends* to achieve the goal of *#stopCar*. An action is a ontological structure relying on the schema provided by DOLCE over processes, consisting of instances of action prototypes (process schemata), connected to action participants as filler of roles within the prototype structure (roles and filler constraints) provided by FrameNet. In our example, *#ProcessSchema1*, connected to the FrameNet frame *Process_stop*, binds the Brother to the *filler* of the *#AgentRole* and Car as the filler of the *#VehicleRole*. Finally, the annotation also includes the intention of Brother to "stop the car "(*#stop Car*, an instance of the *Goal* class) (see below for specifics of goals and intentions) is *obtainedThrough* the action of *#Stopping*, that is connected to the YAGOSUMO commonsense concept *Mechanical Stop*. In [13] we present the validation of the platform through a preliminary test with annotators.

In a dramatic medium, the incidents are arranged on a timeline,

$+_{i=1}^{N} A_i$

Some incident is to be motivated by the achievement of some goal, that is functional to the story advancement. A library of plans, associated with some agent, provides an association of goals and actions (that become actual incidents then) Here, we introduce the plans and the action structure, involving the states that hold before and after an action occurs.

A base plan

$$P[Goal] = +_{i=1}^{M}(PreConditions(A_i)\ A_i\ Effects(A_i))$$

is a sequence of actions (A_i) and states ($PreConditions(A_i)$ and $Effects(A_i)$), where the states are ontological structures similar to actions, ruled by state schemata and connected to participants (+ is the concatenation operator). States can be the pre–conditions and the effects of the actions.

Actions can also be plans themselves; so plans are recursively defined as sequences of (sub)plans and states.

$$P[Goal] = +_{i=1}^{M}(PreConditions(P_i)\ P_i\ Effects(P_i))$$

Pre–conditions can only be mental states, i.e. goals and beliefs. So, given the example of the test, in the short movie "Exit Strategy", there is a scene in which a man, standing near a truck emitting smoke on the wayside, asks people passing by for stopping to help. The base plan identified by the annotator is that the man in stopping people by to ask for help in fixing the truck. The root plan consists of a maintenance goal of going back to work after fixing the truck with the help of somebody who is stopping at him.

The base plan P_{b1}^{Man} has the form:

$P_{b1}^{Man}[G : Man\ wants\ engage\ Driver for(Help)] =$
$B : Man\ believes\ [SOA : Truck\ broken]$
$G : Man\ wants\ [SOA : Man\ engaged\ Driver]$
$A : Man\ askingHelp\ Driver$
$SOA : Man\ confident$

$SOA : Driver\ in\_motion\ near(Man)$
$A : Driver\ stopping\_byMan$
$B : Man\ believes\ [SOA : Man\ engaged\ Driver]$

In this description, the expressions such as "Man engage Driver for(Help)" are short for a ProcessSchema concerning "engage" and connected to a frame in which the participants are the person who engages (Man), the person who is engaged (Driver) and the reason of the engagement (Help). Also, notice that, in this convention, goals (G) require the verb "wants" followed by an action/state with the verb in a neuter form (engage); believes (B) always requires "believes" as verb connecting some agent and the state of affairs believed; state of affairs (SOA) require a past participle ("broken") or an adjective ("confident") applied to some entity; finally, actions are conventionally represented with a verb in the gerund form ("askingHelp").

A higher level plan P_{r11}^{Man}, that includes the base plan P_{b1}^{Man} is:

$P_{r11}^{Man}[PG : Man\ wants\ repair\ Truck] =$
$SOA :\ Truck\ broken$
$P_{b1}^{Man}[AG : Man\ wants\ engage\ Driver for(Help)]$
$SOA : Man\ engaged\ Driver$

$SOA :\ Truck\ broken$
$SOA : Man\ engaged\ Driver$
$P_{b2}^{Man}[PG : Man, Driver\ wants\ fix\ Truck]$
$B : Man, Driver\ believes\ [SOA :\ Truck\ fixed]$
$SOA :\ Truck\ fixed$

In the next section, we show how to augment the timeline representation of incidents to include the mapping between the plan actions and the incidents, in order to provide the input for the joint visualization of timelines and plans.

sectionAugmentation of the Timeline Representation

The aim is to build an intelligent system for establishing a mapping between the actions reported by the plans and the incidents of the timeline and augmenting the timeline with the states that hold between adjacent units, as projected from the plan structure. We have three goals in mind

- through the analysis of the plans, we discover the actions (contained in the plan representation) that match (i.e., motivate) the incidents of the timeline; this is useful for establishing the spatial alignment of the timeline incidents and the plan actions);
- point out successes and failures of characters' behaviors: some plan actions are actually executed (as timeline incidents) and contribute to the plan success, some plan actions can be not executed and the plan fails to accomplish;
- project the states required by the plan, as preconditions or effects of the plan actions, onto the timeline in the places preceding or following the incidents in order to motivate the story advancement through the story states.

In order to implement the intelligent system, we 1) model the timeline and the plans into the ontology, 2) define the incident mapping through SWRL IF–THEN rules, and 3) augment the timeline with states through an off–line algorithm.

Both timeline and plan modeling relies on the generic class *OrderedList*, that represent the positions of the incidents (and actions, respectively) on an ordered list. So, the Timeline class and the Plan class are OrderedList's. An instance of Process or State refers to some position (relation *refersToTimeline*) in the Timeline or in a Plan, respectively.

Based on the representation above, the reasoner infers that some ordered list of incidents in the timeline belongs to some plan (actually, it can happen that some incident is mapped onto more than one plan). The reasoner works with inferences of an ontological nature (for example, the Driver class is a subclass of a PassingByPerson class) and with a SWRL IF–THEN rule that validates the mapping of some incident to some plan action: in particular, the rule tells that, given an incident I in the Timeline and an action A in the plan P,

if
$I.ProcessSchema = A.ProcessSchema$ &
$Role/Filler\ relations\ for\ I\ and\ A\ coincide$
then
$I\ sameAs\ A.$

A similar rule is applied to states. The implementation of the mapping through ontological inferences (such as the subclass relation applied above), supported by the language OWL2 or, in some cases, manually encoded into SWRL IF–THEN rules is an innovative aspect of this work.

Finally, the augmentation of the timeline is implemented through an off–line algorithm that takes as inputs the timeline, the plans, and the incident mapping, and returns as output an OrderedList that contains the incidents of the Timeline, in the same partial order as in the Timeline, interspersed with states (agglomerated into story states) that respect the same partial as reported in the plans. So, if a (plan)state S is a precondition of the action A in the plan P, and the action A is mapped the incident I in the Timeline, then a state S', that is the same as S is inserted in the Timeline before I (and after the incident preceding I in the Timeline). The augmented timeline OrderedList features a total order over incidents and states.

Now we see a brief example of the modeling of plans and timeline. In particular, we see how an incident can be mapped onto two actions of two different plans, respectively. This represents a misunderstanding in an audiovisual fictional tale. In this example, a Man, realizing his Truck is broken, asks the drivers passing by for help. Brother, who is driving by, pretends to be willing to help him by stopping his car, but as soon as the Man checks Brother's will to help, Brother accelerates and departs, thus playing a joke to the Man.

The timeline (check the middle section of figure 2) contains four incidents: the first three are originated by agents' actions, the fourth is an event (I_05_2: Girlfriend awakening in(Car)). The three actional incidents are:

- I_03: Man asking_help
- I_04: Brother stopping Car
- I_05_1: Brother leaving with(Car)

We have two plans. One is the plan P_{b5}^{Man}, which collapses two of the plans above for simplicity of exposition:

$P_{b1}^{Man}[AG : Man\ fix\ Truck\ with(Driver)] =$

$B :\ Man\ believes\ [SOA : Truck\ broken]$
$G :\ Man\ wants\ [SOA :\ Man\ engaged\ Driver]$
$A :\ Man\ askingHelp\ Driver$
$SOA :\ Man\ confident$

$SOA : Driver\ in\_motion\ near(Man)$
$A :\ Driver\ stopping\ Car$
$B :\ Man\ believes\ [SOA :\ Man\ engaged\ Driver]$

$SOA :\ Driver\ willing$
$A :\ Man, Driver\ fixing\ Truck$
$SOA :\ Truck\ fixed$

The other plan concerns the Brother, $P_{b2}^{Brother}$:

$P_{b2}^{Brother}[PG :\ Brother\ making\_joke\ to(Man)\ with(Car)] =$

$B :\ Brother\ believes\ [SOA :\ Truck\ broken]$
$A :\ Brother\ stopping\ Car$
$SOA :\ Car\ stopped$

$B :\ Brother\ believes\ [SOA :\ Man\ victim]$
$A :\ Brother\ leaving\ with(Car)$
$B :\ Man\ believes\ [SOA :\ Brother\ unwilling]$

The incident/action mappings are the following. The incident "I_03: Man asking_help" is mapped onto the action *askingHelp(Man, Driver)*: in this case, the action description adds the receiver of the asking action, namely the Driver, to establish the sameAs relation; the incident "I_04: Brother stopping Car" is mapped onto the action *stopping(Driver, Car)*, thus equalizing Brother and Driver; finally, "I_05_1: Brother leaving with(Car)" is immediately mapped onto *Brother leaving with(Car)*. Notice that in this example we are using instantiated actions, with variables already replaced by the actual instances that are used in the mapping. The use of generalized actions, which is certainly advisable for plan definition, is not in the scope of this paper.

A plan participates to the mapping and the augmentation of the timeline when the order of the incidents on the timeline respects the order of the mapped actions in the plan. In our example, the incident "I_03: Man asking_help" maps onto the action $A :\ Man\ askingHelp\ Driver$, followed by the incident "I_04: Brother stopping Car" mapping onto the action $A :\ Driver\ stopping\ Car$, so the plan P_{b1}^{Man} can participate to mapping (notice that the last part of the plan is not mapped then). Similarly, "I_04: Brother stopping Car" maps onto the action $A :\ Brother\ stopping\ Car$, followed by "I_05_1: Brother leaving with(Car)" mapping onto $A :\ Brother\ leaving\ with(Car)$, so the plan $P_{b2}^{Brother}$ can also participate to mapping.

If the sequence of incidents does not respect the order exhibited by the mapping actions in some plan, that plan is not activated for contribution to mapping.

Once we have identified the incident–action mapping and the plans that contribute to such mapping, we augment the timeline with the states that hold between adjacent incidents on the timeline. States are taken from the pre–conditions and the effects that are associated with the actions in the plans and are employed to augment the timeline that will be visualized by the tool. So, in the case of the triplet

- $B :\ Brother\ believes\ [SOA :\ Truck\ broken]$
- $A :\ Brother\ stopping\ Car$
- $SOA :\ Car\ stopped$

with the action mapped onto the incident "I_04: Brother stopping Car", we have that the state $B :\ Brother\ believes\ [SOA :\ Truck\ broken]$ will precede the incident, while the state $SOA :\ Car\ stopped$ will follow the incident.

In order to implement the timeline augmentation, we have devised the following general rule:

$Dynamics(?x),$
$hasStateEffect(?x, ?e),$
$hasStatePrecondition(?x, ?p),$
$refersToTimeline(?x, ?t)$
$- >$
$hasDynamics(?x, ?p),$
$hasDynamics(?x, ?e),$
$refersToTimeline(?p, ?t),$
$refersToTimeline(?e, ?t)$

This rule states that: if x is an instance of the class Dynamics and its position on the timeline is known (y), and x has an effect p and a precondition z, then the preconditions and effects of x will be assigned to the same position y in the timeline. It is the relations "hasStatePrecondition" and "hasStateEffect" that allows the positioning before or after the incident, exactly.

III. CADMOS VISUALIZATION TOOL

In this section we describe the design, both interface and interaction, and the implementation of the visualization tool. The visualization concerns multiple trees of characters' intentions (or plans), possibly arranged hierarchically on a tree that spans a timeline of events.

The whole visualization space is split into three areas (refer to figure 2): the Agents area (top), where the characters involved (called *agents* in project CADMOS) are listed and can be selected for partial visualizations, the Timeline area, where the augmented timeline is displayed with the incidents (grouped in Units) and the states, grouped in StoryStates, the Plans area, where the plans spanning the timeline incidents are displayed, with actions/states aligned with the mapped incidents/states, or possibly with higher plans aligned with lower plans.

Each narrative incident or state is represented by a box, green color for actions A and events E (lighter green in this case), red color for states. Boxes filled of white in the Plans area (P) means have not been mapped yet to some element in the timeline, but the plan is activated because some of the plan elements (actions or states) have been mapped. Finally, the boxes filled with white color and barred diagonally means have not been realized in the Timeline, thus the plan failed.

All the incidents or states in the timeline have occurred in the plot realization. The timeline incidents pivot the hor-

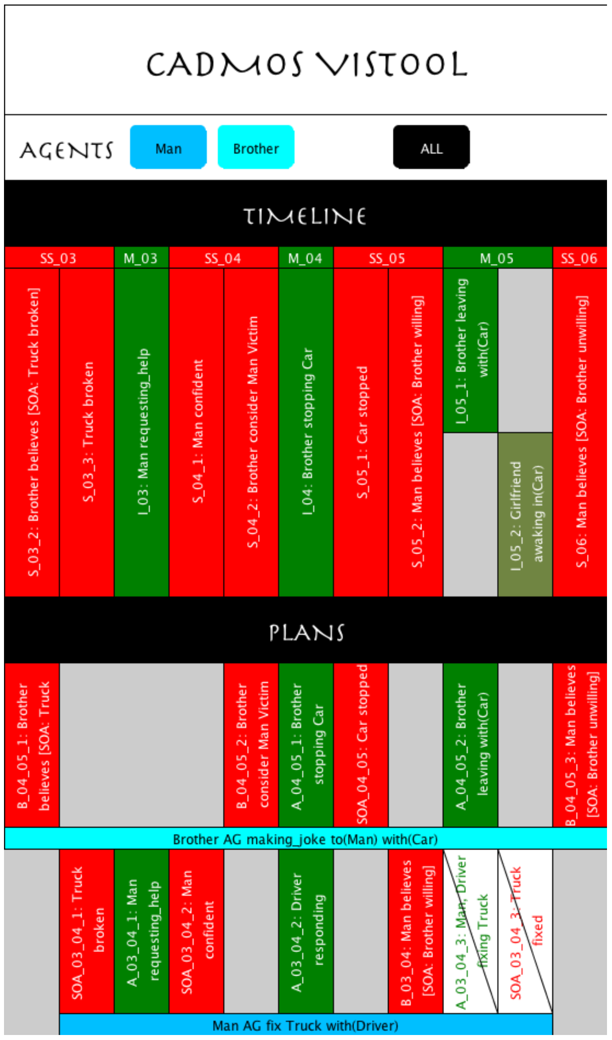
CADMOS VISTOOL
AGENTS
Man
Brother
ALL
TIMELINE
SS_03
M_03
SS_04
M_04
SS_05
M_05
SS_06
S_03_2: Brother believes [SOA: Truck broken]
S_03_3: Truck broken
I_03: Man requesting_help
S_04_1: Man confident
S_04_2: Brother consider Man Victim
I_04: Brother stopping Car
S_05_1: Car stopped
S_05_2: Man believes [SOA: Brother willing]
I_05_1: Brother leaving with(Car)
I_05_2: Girlfriend awaking in(Car)
S_06: Man believes [SOA: Brother unwilling]
PLANS
B_04_05_1: Brother believes [SOA: Truck
B_04_05_2: Brother consider Man Victim
A_04_05_1: Brother stopping Car
SOA_04_05: Car stopped
A_04_05_2: Brother leaving with(Car)
B_04_05_3: Man believes [SOA: Brother unwilling]
Brother AG making_joke to(Man) with(Car)
SOA_03_04_1: Truck broken
A_03_04_1: Man requesting_help
SOA_03_04_2: Man confident
A_03_04_2: Driver responding
B_03_04: Man believes [SOA: Brother willing]
A_03_04_3: Man, Driver fixing Truck
SOA_03_04_3: Truck fixed
Man AG fix Truck with(Driver)

Fig. 2. (c) Augmenting timeline with states projected from plans.

izontal alignment: each realized plan action is aligned with the matching timeline incident; at the same time states of the plans are propagated to the timeline to represent the story state between adjacent units. The incidents that occur in a unit are considered in parallel, though we decided to assign them an individual position to allow for a visible alignment with the plan action. The plan label is an horizontal box that spans all the states and actions that belong to it.

In figure 2 there is the visualization of the excerpt of "Exit Strategy" described above.

The visualization algorithm proceeds left to right by following the mapping between incidents and plan actions. It assumes the timeline distribution of the states and incidents over the x axis as fixed and aligns the plan actions and consequently the precondition and effect states as a consequence. The plan hierarchy is built downwards, so higher layers will be lower in the visualization. Plans fill lines close to the timeline first and as soon as the alignment does not allow to fit other plans, the algorithm goes lower in the visualization space, proceeding through layers.

IV. Evaluating the Effectiveness of the Visualization Tool in Teaching Drama Analysis

The annotation was conducted on the short film *Exit Strategy* produced by Lumiq Studios as part of project CADMOS. We conducted a preliminary annotation of the screenplay as follows. We segmented the storyline into scenes following the author's description, unless the scenes were already split for mere camera editing reason (e.g., the screenplay indicates that the action has to be seen from several angles and insert a "cut on" in the middle of the action). Then, for each unit we selected one character that seems to show an intentional behavior. For each behavior, we introduced a *plan*. For intentional behavior we intend that the character is not simply reacting to the situation but is the initiator of the action. Therefore we do not describe neither the character's emotions nor the event that happens by chance. First, we name a plan and its agent; then we list the preconditional states (material or mental) on which it is grounded, the action planned, and the consequent state expected. Each of these is marked as *achieved* or *unachieved* value. By focusing on the deliberation, this annotation stresses the credibility and motivation of character's behavior.

Once we have a hierarchy of plans, we annotate the video using the same actions, events, states, beliefs, and we render a sequence of timeline elements as perceived from the audience point of view. Incidentally, we discover the elements of the cinematic text that were not rendered by the screenplay annotation. For example, an incident that was not in any plan (because not belonging to any deliberation), could be annotated anywhere in the timeline.

Next, the visualization tool allows us to match the two annotations. Here we see how the timeline corresponds to the constant re-planning of the character as the actions or states they figured in the deliberation fail. Even if the timeline is grounded onto the deliberations of the characters, it needs to count for event and emotion to get its consistency. This led to a specific question about the difference between text and mise–en–scène so ubiquitous in the literature as the difference between drama and performance. There are actions that were implicit in the screenplay so that the annotator had not to annotate them. On the contrary, the actions are stressed in the shooting, rendered in such a way that the timeline annotator had to describe them. For example, this is the case where the man, in need of help for his truck, is deceived by the brother who drives away laughing (see previous sections for plans). Here the shooting and the final editing focus on the deluded man: a state and an action that were given for granted in the script. This leads to a mismatch between the action listed in the plans, and those listed in the timeline, with the mise–en–scène creating a new specific bit of meaning adding to the given script.

The character's behavior in the script is described diachronically and such are the plans into the annotation. The timeline needs to account for the synchronic delivering of the character's actions such has to select the bit of meaning that are the most relevant in each scene for the narrative to progress.

For example, there is a fast and intense scene in which the burglars assault the group of friends. Here the complex clash of behaviors is rendered into a stratification of actions, well rendered in the visualization, on which only few are directly matched onto the timeline. This means that the mise–en–scène has to select bits of meaning that can be synchronic and account for the diverse intentions.

In teaching drama, it is relevant to describe the dramaturgy of the performance. Our example shows that the plan list can be almost completely mapped onto the timeline, hence showing that the narrative text of dramatic medium is bounded to character's deliberation. In terms of learning about drama structure and meaning, our visualization helps to bridge the gap between the descriptions of the script and of the performance, respectively, and shows the interventions of the latter in terms of dramaturgy. In fact, performance not only adds to the temporal and spatial quality to drama, but also re-shapes the dramaturgy of the script. Therefore, it is relevant to describe the dramaturgy of the performance.

Nowadays drama courses tend to switch from literary to actional qualities. This means that the text is increasingly intended as an incident design (either on stage or on screen). The visualization system allows the teachers to clearly stress the structural elements in the text that are linked to the performance, i.e., it shows the continuity between event design and event performance. The example we have used shows that the plan list can be almost completely mapped onto the timeline, hence can help the class to understand that the narrative text of dramatic medium is bounded to the character's deliberation. This can be used to teach the students how to read the *characters behaviors*. Furthermore, the stratification of agent's plan, as seen in the assault scene, and its mapping onto the timeline, helps the teacher to visualize the *orchestration of conflicts* and their synchronic execution. The visualization of the failed plans, causing a re-deliberation, is a clear indication of the *characters' change* as a key figure into the emotional engagement of the audience. In general, in learning about drama structure and meaning, our visualization helps to bridge the gap among the description of script and of performance, and shows the interventions of the latter in terms of dramaturgy. Nevertheless, our visualization could be more effective if the timeline were expressed also in terms of frames and timecode to give teachers and students a more direct access to the audiovisual document.

V. Conclusion

This paper has presented a tool for the visualization of the characters' intentions in a narrative plot. Character's intentions form multiple trees that span a timeline of incidents and the tool is an interactive system for the tree visualization and the selection of the characters one by one. The system is able to build the mapping between a library of plans and the timeline of incidents, and to visualize the contributions of the several characters' intentions to the whole plot.

The system relies on an ontology of drama and builds upon the unrestricted annotation provided by narrative enthusiasts and media students. The system was tested on the analysis and exposition of the case of a short movie for testing the applicability of the annotation to media production. We hope that such a tool can be useful for media scholars and analysts, and it is susceptible of applicative scenarios within the media industry. Though oriented and tested to the didactics of drama structure our system can be applied to the analysis of news stories, blog entries, or the fruition of cultural heritage through web and mobile applications (see, e.g., [14]).

Acknowledgment

The work presented has been partially supported by project CADMOS (Character–centred Annotation of Dramatic Media Objects - http://cadmos.di.unito.it), funded by Regione Piemonte, Innovation Hub for in Multimedia and Digital Creativity, 2010-2012, POR-FESR 07-13. We thank David Pugliese, Carmi Terzulli and Dario Mimmo for their annotation work.

References

[1] S. Mamber, "Narrative mapping," in *New Media: Theories and Practices of Intertextuality*, A. Everett and J. Caldwell, Eds. Routledge, 2003, pp. 145–158.

[2] D. Fisher, A. Hoff, G. G. Robertson, and M. Hurst, "Narratives: A visualization to track narrative events as they develop." in *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (IEEE VAST 2008)*, vol. 01, Columbus, Ohio, USA, 2008.

[3] A. Baikadi, J. Goth, C. M. Mitchell, E. Y. Ha, B. W. Mott, and J. C. Lester, "Towards a computational model of narrative visualization," in *AIIDE 2011 - AAAI Workshops at the Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2011, pp. 2–9.

[4] M. Esslin, *The Field of Drama*. London: Methuen, 1988 (1987).

[5] M. Graham and J. B. Kennedy, "A survey of multiple tree visualisation," *Information Visualization*, vol. 9, no. 4, pp. 235–252, 2010.

[6] S. K. Card, B. Suh, B. Pendleton, J. Heer, and J. W. Bodnar, "Timetree: Exploring time changing hierarchies," in *IEEE Symposium on Visual Analytics Science and Technology (VAST)*.

[7] M. Cataldi, R. Damiano, V. Lombardo, A. Pizzo, and D. Sergi, "Integrating commonsense knowledge into the semantic annotation of narrative media objects," *AI* IA 2011: Artificial Intelligence Around Man and Beyond*, pp. 312–323, 2011.

[8] A. Pease, I. Niles, and J. Li, "The suggested upper merged ontology: A large ontology for the semantic web and its applications," in *In Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web*, 2002.

[9] F. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 697–706.

[10] G. De Melo, F. Suchanek, and A. Pease, "Integrating yago into the suggested upper merged ontology," in *Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on*, vol. 1. IEEE, 2008, pp. 190–193.

[11] C. Baker, C. Fillmore, and J. Lowe, "The berkeley framenet project," in *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 1998, pp. 86–90.

[12] A. Gangemi and V. Presutti, "Ontology design patterns," *Handbook on Ontologies*, pp. 221–243, 2009.

[13] V. Lombardo and R. Damiano, "Commonsense knowledge for the collection of ground truth data on semantic descriptors," in *Proceedings of the 2012 IEEE International Symposium on Multimedia (ISM 2012)*. IEEE Computer Society, 2012, pp. 78–83.

[14] S. Andolina, D. Pirrone, G. Russo, S. Sorce, and A. Gentile, "Exploitation of mobile access to context-based information in cultural heritage fruition," in *Seventh International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA)*, 2012, pp. 322–328.