TECFA

Technologies de Formation et Apprentissage
Educational Technology Unit

TECFA - FPSE
Bd du Pont-d'Arve, 40
Université de Genève
CH 1211 Genève 4
SWITZERLAND

Performative Structures for Interactive Narrative: An authored-centred approach

Nicolas Szilas
TECFA
nicolas.szilas@unige.ch

Urs Richle
TECFA
urs.richle@unige.ch

Paolo Petta
Austrian Research Institute for Artificial Intelligence, OFAI
paolo.petta@ofai.at

Technical Report TECFA-12-1

Jan. 2012

Research funded by the European Commission under grant agreement IRIS (FP7-ICT-231824)
and the Swiss National Science Foundation (J. Dumas and N. Szilas, principal investigators).
The Austrian Research Institute for Artificial Intelligence (OFAI) is supported by the Austrian Federal Ministry for Transport, Innovation, and Technology.

UNIVERSITÉ DE GENÈVE
FACULTÉ DE PSYCHOLOGIE ET DES SCIENCES DE L'ÉDUCATION

ABSTRACT
The field of Interactive Narrative is promoting a growing body of research in Artificial Intelligence and Autonomous Agents. Interactive storytelling systems have to embed computational models of narrative; these are usually based on existing research in the computational disciplines (e.g., planning, dialog-modeling, etc.) However, when an agent participates in a narrative, the key perspective on its actions is shifted away from the immediate and situated contexts typically considered in application domains such as robotics and human-computer interaction. We propose that by representing agents' actions in a more narrative-oriented formalism we can:

- Simplify the modelling of interesting narrative phenomena and thus increase the quality of generated/interacted stories.
- Ease the authoring process, which is broadly recognised to constitute a critical issue in the field.

The paper focuses on what we call "Performative Structures", a fundamental component that is shared by most Interactive Narrative systems. It consists of a set of agents' tasks and goals, along with their mutual relationships. In this paper, we lay the foundation of a new performative structure model called PS-101, which is based on four first-class elements: goals, tasks, obstacles and side effects, along with nine types of relations among them. These elements and relations constitute the language that is handled when building a performative structure for a given interactive narrative. As one of the main contributions, an author-centred visual representation for these elements and relations will be presented in detail.

1. INTRODUCTION
For two decades, the field of Interactive Narrative (as well as related fields termed e.g. Interactive Drama or Interactive Storytelling) has been motivating a growing body of research in Artificial Intelligence [1,3,6,7,10,17,18,24,25,29,34,38]. Most of the Interactive Narrative systems employ virtual agents, endowed with varying levels of autonomy, ranging from purely autonomous agents in narrative approaches based on narrative emergence [1,21] over behavioural autonomous agents controlled by a drama manager [3,18] to more centralized computational models of narrative [29,39]. Most of these approaches attempt to solve the difficult issue of allowing strong user intervention in the unfolding of story events while safeguarding overall narrative coherence and quality, as expressed by an author. It is difficult to provide a unified view on the different approaches that have been followed to achieve this goal, as they are particularly diverse and the field in itself remains largely unexplored. Still, a common feature is for narrative progression to be represented in a state-oriented view in terms of *goals* and *tasks,* a goal being a specific state in the fictional word that one wishes to reach, and a task (also termed action or operator) being a transformation of the fictional world from one state to another. Goals and tasks not only constitute common concepts in Artificial Intelligence (AI) but often are also taken to correspond to fundamental narrative entities. For example, the notion of *goal* can be taken to correspond to one of the six main *actants* in structuralist narratology [13] and is likewise seen as basic ingredient of narrative in dramaturgy and screenwriting [11,37]. Likewise, a *task* means the narrative transformation performed by an *agent* in Bremond's theory [5], and also corresponds to the narrative concept of *verb* in Todorov's theory [36]. Goals and tasks are organized in different ways across different approaches to interactive narrative. We term this organization a *performative structure*. This organization serves the purpose of computations performed by, for example, planning [39], rule-based action selection [24], or subgoaling [32].

These efforts on performative structures tend to place a focus on single phenomena of experiential interest ("causality of actions", "emotional expressivity", "suspense") within which they are guided by considerations of feasibility, computational complexity, performance, or scalability. This collection of requirements of the kind, "*what needs to be specified (explicitly) in order to use which technology to achieve what effect*" on the one hand constitutes an essential "bottom-up" effort that clarifies the relation of Interactive Narratives to other media such as classic drama, movies, or TV soaps in terms of commonalities and important differences. From a "top-down" authoring point of view, this approach leads to a range of questions to be addressed, such as the relation/match of the first-class entities of the computational models to narrative counterpart and issues of usability.

As another example, an important thrust in research on communication within multi-agent systems adopted speech act theory as foundation [12]. This led to the specification and implementation of a collection of individual speech acts which were then offered to authors as building blocks out of which to assemble all communication activity. Only then was it more broadly realised that rather than considering individual speech acts, whole conversations (carried out for application-specific purposes) most often formed more useful and appropriate basic conceptual entities for the design of multi-agent applications [20]. In this sense, the additional degree of freedom apparently offered by the possibility to assemble largely unconstrained sequences of speech acts was not at all advantageous for authors.

This paper promotes this rationale for performative structuring. In our effort, we try to address the questions introduced above by paying explicit attention to the following two viewpoints:

- *Narratological perspective*: Performative structures based directly on narrative principles offer additional options to existing systems and motivate the design of novel interactive narrative mechanisms (e.g. by providing articulated context-dependent semantics for representations). Narrative principles already used in Interactive Narrative and Story Generation include ethical conflict [24,29], obstacles [30], dilemmas [2], or the actant model [14].
- *Authorability perspective*: Authoring has been identified as a key issue in Interactive Narrative [25,27,33]. Motivated by considerations such as the above questions, we aim to emphasise the authors' ability to *use* the performative structure as accessible and expressive material. It is our expectation that this should lead to more artistic and novel work that exploits

and expresses more of the potential of Interactive Narrative. An additional derived evaluation measure will be given by the number of complete stories produced.

Combined, these two viewpoints constitute what we call a narrative-oriented approach. In the performative structure proposed, authorability considerations are thus favoured over (or at least put next to) qualities usually focused on in AI, such as computational complexity or representational power. Alongside, empirical evidence from authoring with AI-based formalisms has led to simplifications, e.g. in the way pre- and postconditions of tasks are managed. Even so, this paper is not a step back to story graphs and branching structures: The principle of *generativity* remains central for the type of Interactive Narrative we are aiming for [32] and accordingly is supported by the performative structure. Thus, focusing more on usability encourages the exploration of new formalisms for narrative agents without sacrificing computational qualities. Historically, such orientation has proven useful: Petri Nets were invented in part with the requirement of facilitating the use of computing systems [4].

This paper focuses on the performative structure because it constitutes a key component for a large class of systems based on computational narrative. For example, planning can make use of such a performative structure, but goals can also be part of an exchange of information between two agents, such as when one agent is pursuing a specific goal or requests to delegate this goal. Even though such mechanisms will not be discussed in depth, it should be kept in mind that they can, and indeed should be, used in conjunction with the performative structure PS-101 we are describing next.

2. THE PERFORMATIVE STRUCTURE PS-101

2.1 Derivation Method

Formal characterization of a performative structure is based on the computational structures constituting the building blocks for Interactive Narrative models. Following the constraints exposed in the introduction, these *first-class elements* should be narrative-related, author-friendly, and relevant in terms of story generation and interaction.

There is no unique solution for these design requirements, authoring-related constraints included. It is also important to note that while direct involvement of authors in the derivation process is a straightforward necessity, they themselves cannot be expected to be able to express directly, explicitly, and exhaustively what is meaningful for them, either–in particular not for the kind of novel interactive narratives we are aiming for. In other words, we must not expect authors to e.g. spontaneously assert: *"I need the concept of postcondition to write an interactive narrative"*. Authors *can,* however, intervene in two manners:

- They can inform us after the development of a particular system which structures did and did not work for them.
- While appropriating the formalism, they may spontaneously re-interpret it in a way that is more suitable to them. Such reinterpretation can inspire new formalisms and visualisations.

Thus, iterative design of an interactive narrative engine appears to be well-suited to approach the satisfying of the authorability constraint. Starting with the IDtension narrative engine [29-31], such iterating has taken place, in close and long-term collaboration with an author; resulting in the performative structure described here. We start by taking a closer look at the four first-class elements in PS-101: goals, tasks, obstacles, and side-effects. Note that as explained in the introduction, we leave it open how a specific algorithm makes use of elements of the Performative Structure.

2.2 PS-101 First-class elements

2.2.1 Goals

In PS-101, *goals* represent what characters want to achieve as a state of affair in the fictional world. Goals are expressed simply by stative verbs such as *be*, *have*, *love*, or *believe*. To increase generativity, goals may include parameters: named terms left undefined at authoring time and instantiated at execution time with concrete *entities*: characters, objects, or places. Once instantiated (see 2.3.7, "Chaining"), goals are either *active* (currently pursued by the character who has it), or *reached*.

It is useful to use predicates for the notation of goals. As an example for the notation of a *simple* goal (i.e. without parameters) the goal of being rich can be stated as *be_rich*. In goal specifications, parameter names are preceded by question marks. For example, *have(?object)* expresses the goal to have (possess, hold, …) the *entity* referred to by the variable named "object".

As discussed in the introduction, the concept of a goal is used not only in psychology and AI (notions related to PS-101 include "*achievement goal*" and "*intention*" for actively pursued goal instances) but also in narrative theories (if termed differently). This is not only due to the fact that narrative is about people and people have goals, but also because classic narrative is structured around an initial imbalance or problem to be solved by the protagonist. This parallel between narrative structure and problem solving has inspired some story generation techniques [19].

2.2.2 Tasks

In PS-101, *tasks* are concrete actions characters can perform to reach a goal. Tasks are expressed by dynamic verbs such as *eat*, *steal*, *offer*, or *read*. A task is always associated to a single goal, which is reached when the task is performed successfully. The same goal may be reached by multiple tasks.

Such explicit association between goal and task has been chosen for simplicity, as the common alternative requiring specification of explicit pre- and post-conditions for tasks (STRIPS-style) was found to be less natural for authors. Our approach relates to associations between tasks and compound tasks as found in Hierarchical Task Networks [23] and task decomposition models such as TÆMS structures [14,37], which additionally support explicit hard ("hinders", "enables") and soft ("facilitates", etc.) interrelationships between tasks and "quality accumulation functions" describing how variants of sub-steps contribute to an overall quality of task execution. In this sense, TÆMS structures have further similarities to whole configurations of goals, tasks, obstacles, and side effects in PS-101.

Tasks also can have parameters. For example, *steal(?actor, ?object, ?owner)* may represent the stealing of the object referred to by the variable named *?object* by the actor *?actor*.

The author specifies whether task parameters are assigned explicitly or implicitly. In an *explicit* specification, each task parameter is matched with a goal parameter or a static value. In *implicit* parameter specification, the author uses a set of con-straints on the parameters. In PS-101, two constraints are introduced: *all* and *different*. The "all" constraint allows the parameter to be filled by any existing entity. With the "different" constraint, the parameter can take on any entity that differs from a given entity. Unless specified otherwise, a task inherits all parameters of the goal it is associated to. Task instantiation may further depend on *task conditions* for parameters.

2.2.3 Obstacles

Obstacles represent *failure* events that can happen when a task is attempted. An obstacle prevents the character who attempted the task to achieve the goal the task is associated to. The concept of obstacle is directly imported from dramaturgy and screen writing [9,36] even if terminology differs. It is readily understood and appreciated by authors. It is particularly relevant in the context of interactive narrative (as opposed to story generation), since it introduces the essential quality of non-determinisim.

The concept of "obstacle" is already used in IDtension [31]. Obstacles are also similar to the concept of planning failures. In the domain of comic situations, failures are due to non-verified executability conditions [7]. The intervention mechanism in the Mimesis architecture uses failures to modify action execution so as to preserve global narrative constraints [22].

An obstacle is always associated to a task. A task can have several obstacles associated to it, in which case any of these obstacles can potentially *trigger* during task execution. Each obstacle is assigned a probability of triggering, also called a *risk*. If an obstacle's *triggering conditions* are met, its risk is used to decide whether to actually trigger it. An obstacle is *on* when its triggering conditions are met, and *off* otherwise. If not explicitly specified otherwise, the risk of an obstacle is 1.0, meaning that it will unconditionally be triggered whenever it is *on*. Note that the narrative engine may take a global decision to change the probabilities of obstacles that are *on* for reasons other than the performative structure. (See for exemple the intervention algorithm mentioned above [22].)

Examples of obstacle notations: *door_locked* – associated to a task *open_door*; *get_angrier(?angryPerson)* – associated to a task *calm_down(?angryPerson)*.

2.2.4 Side effects

Side effects also occur during the execution of a task. However, while obstacles cause a task to fail, side effects do not influence the reaching of the goal the task is associated to. Side effects have other positive or negative consequences *outside* of this goal, as detailed in the discussion of the relational elements of PS-101.

Unlike obstacles, the *triggering probability* of a side effect is not called a risk since a side effect may have positive consequences, such as facilitating achievement of another goal or removing the conditions of an obstacle that is *on* before it is triggered.

Side effects are also narratively meaningful. When not known in advance by the character, they correspond to involuntary actions, as described by Bremond: An agent undertakes a task but performs at the same time an "involuntary action" [5, p.237].

2.3 Relational elements

The presentation of first-class elements of PS-101 did not include any explicit formal notion of *state*. This is motivated by our observation that this concept–when represented e.g. in terms of first-order logic predicates–is difficult to adopt for users lacking formal training. Instead, PS-101 makes use of *relations* between these elements. These relations describe in an abstract way the potential dynamics of the performative structure. Relations are *oriented* from a *source* element to a *target* element.

Relations are visualized graphically by arrows from source to target. Generally speaking, such visualization is intuitive and is widely used in Computer Science, in theories as varied as Graph Theory, Petri Nets, or Bayesian Networks. We observed on a previous narrative model lacking such relational elements that the author would spontaneously add links in the design documents whenever they could be inferred from formal specifications that included pre-condition and post-conditions [8]. The semantics of these spontaneously denoted links however was not uniform. PS-101 aims to formalize the meaning of such links and to systematize their use.

PS-101 currently comprises nine relation types. This is not meant to constitute a final and canonical description of narrative relations; future revisions of the model may include additions.

2.3.1 Subgoaling (sub)

A *subgoaling* relation links an obstacle, termed the *triggering obstacle*, to a goal, termed the *subgoal*. When an obstacle is triggered, it triggers the subgoal associated to it. The subgoal then needs to be reached first in order to achieve the character's overall goal. Conversely, if the subgoal is not reached, then the obstacle remains *on* (and will trigger with the probability specified by the author-defined *risk*). For example, an obstacle "the door is locked" may trigger a subgoal "have the key". If the narrative engine supports it, a subgoal may already be triggered when the obstacle relating to it is merely *anticipated*.

In case the subgoal is parameterized, the subgoaling link must specify how these parameters are to be instantiated. There are three ways of instantiation.

1) Some parameters are instantiated with values inherited from the triggering obstacle.
2) Some parameters are instantiated with static entities: that is, although this subgoal has a parameter, in this context it is specified that it will take a given entity as value.
3) Some parameters are instantiated with default values. For example, the parameter *?actor* in a subgoal automatically takes the value of the eponymous parameter of the triggering obstacle.

The cases 1) and 2) are specified within the subgoaling relation.

Subgoals could be subdivided further into two types, depending on whether the actor pursuing the subgoal is identical to the actor pursuing the goal related to the triggering obstacle or not. The former case corresponds to the usual intra-agent sub-goaling mechanism while the latter could be termed a *co-goal*, with another character helping the first to remove the obstacle. However, this distinction is not modelled within PS-101, both cases being handled by the single subgoaling mechanism.

2.3.2 Counter-goaling (cts and ctg)

A *counter-goaling* relation links a goal (termed the *counter-goal*) to an obstacle (termed the *blocked obstacle*). Whenever the counter-goal is reached, the associated blocked obstacle becomes *on*. If the overall goal (hindered by the blocked obstacle) and the counter-goal belong to two different characters, it creates a conflict of interest or a voluntary obstructing of one character by the other.

PS-101 distinguishes two types of counter-goaling relations. In the first type, named *simple counter-goaling (cts)*, the obstacle exists (is instantiated) independently from the instantiation (i.e., active pursuit) of the counter-goal. In the second type, named *generative counter-goaling (ctg)*, the obstacle is instantiated only when the counter-goal is instantiated.

Like subgoals, counter-goals can have parameters, these are instantiated in the same manner as for the subgoals.

2.3.3 Obstruction (obs and obg)

An *obstruction* relation is a relation between a side effect and an obstacle. As soon as the side effect is met (triggered), the obstacle is set to *on*. Obstructions can be used to model direct causality between events. Similarly to counter-goaling, two types of obstruction relations are distinguished in PS-101. In *simple obstruction* (*obs*), the obstacle exists (is instantiated) independently from the existence of the side effect. In *generative obstruction* (*obg*), the obstacle is instantiated only when the side effect is instantiated. (see Section 2.5.1).

Note that there is no equivalent mechanism to subgoaling that would trigger a goal to avoid the side effect. By design, side effects are not changeable. Narratively speaking, their triggering is "in hand of destiny".

2.3.4 Clearing (cle)

A *clearing* relation also holds between a side effect and an obstacle. It allows to model that a side effect clears the obstacle. In this sense, it is the opposite of the obstruction relation and corresponds to a positive consequence of a side effect. As soon as the side effect is triggered, the obstacle is cleared (set to *off*) and no longer hinders the execution of the task it is associated to.

2.3.5 Needing (nee)

A *needing* relation is a relation between a side effect and a goal. It means that as soon as the side effect is triggered, a new goal is created. It serves a similar function as the *obstruction* relation, but in an indirect manner, via a goal. Furthermore, it enables the creation of a new goal *after* the triggering of a side effect. The distinction of obstruction and needing in PS-101 is also the result of empirical observations.

2.3.6 Solving (slv)

A *solving* relation is another relation between a side effect and a goal. It means that as soon as the side effect is triggered, the target goal, if activated, is reached. It models a positive side effect of achieving a first goal with a second goal "in the pipeline". This relation is opposite to the *needing* relation.

2.3.7 Chaining (chn)

Chaining is a relation between two goals. When a first goal is reached, the second goal is activated. It is a simple relation that (re-)introduces some linearity, but at the higher level of goals rather than of tasks. Note that for a character under user control it may be preferable to propose activation of the second goal as an option to the user, rather than activating it automatically.

2.4 Rules of execution

The authoring process leads to a declarative narrative structure, graphically represented by a graph relating/connecting the narrative elements. This narrative structure is abstract, in the sense that in most cases it does not represent particular narrative events nor specific causal relations between events: It describes potential events and their interrelations. According to the execution (including the end-user's interaction), these potential events materialize as different concrete events, in differing order.

2.4.1 Global execution

Execution of an abstract PS-101 performative structure incrementally materialises a concrete *execution structure* that corresponds to the current state of affairs of the story at any given moment during the execution. This execution structure is built by instantiating some goals in the abstract structure, along with the associated tasks, obstacles, side effects, and relations. Note that a single abstract narrative element (a goal, a task) can produce several instantiations for the same execution structure, depending on the parameters.

The execution starts with some *initial conditions*, i.e., one or more instantiated active goals. These instantiated goals can be defined by the author, or governed by some *initialization rules*.

Each time the execution structure is modified, any entailed maintenance computations are automatically performed on it:

- When a goal is activated or deactivated, associated tasks are added to or deleted from the execution structure.
- When a task is added: associated obstacles are added, except for obstacles linked to by generative obstruction or generative counter-goaling relations; associated side effects are added.
- When a task is deleted, associated obstacles and side effects are deleted.
- When an obstacle or a side effect is added or a goal is activated, a check is performed whether any incoming or outgoing relations can be instantiated, due to already instantiated obstacles or side effects or due to an active goal.
- When a side effect is added, a check is performed whether any obstacle can be instantiated, based on a generative obstruction relation outgoing from the obstacle. Such an obstacle can be instantiated if its task is already instantiated.
- When a goal is activated, a check is performed whether any obstacle can be instantiated, according to any generative counter-goaling relation originating from the goal. Such an obstacle can be instantiated if its task is already instantiated.
- When an obstacle or a side effect is deleted, all associated incoming or outgoing relations are deleted.
- When an obstacle is triggered, subgoaling relations and corresponding subgoals are instantiated or activated.
- Similarly, when a side effect is triggered, needing relations and corresponding goals are instantiated or activated.
- When a goal is reached, chaining relations are instantiated and corresponding goals are instantiated/activated.

Once these maintenance computations have been performed, it is the role of the narrative engine that makes use of the performative structure to calculate and perform a certain number of *narrative actions*. In particular, the narrative engine triggers the execution of some tasks or lets the end-user perform some tasks. These tasks will either succeed or fail. Failure corresponds to the triggering of an obstacle (see below). Success deactivates the goal(s) to which the task(s) are associated, and their status is changed to *reached*. These modifications of the execution structure cause further maintenance changes, according to the above rules.

2.4.2 *Obstacle and side effect triggering: The simple case*

This simple case concerns the situation where the executed task contains only one obstacle or side effect, and this element contains a maximum of one outgoing or incoming relation.

For obstacles, it first needs to be examined whether the obstacle is *on* or *off*:

- If the obstacle has no relation connected to it, then the obstacle is *on*.
- If there is a subgoaling relation, the obstacle is *on* if and only if the subgoal is not reached.
- If there is a counter-goaling relation, then the obstacle is *on* if and only if the counter-goal is reached.
- If there is an obstruction relation, then the obstacle is *on* if and only if the side effect has been triggered.
- If there is a clearing relation, then the obstacle is *on* if and only if the side effect has not been triggered.

Second, it must be decided whether the obstacle will trigger or not. If the obstacle is *off*, the obstacle will not trigger. If it is *on*, triggering depends on the associated risk, in a probabilistic manner if the risk is below 1.0.

For side effects, only the second step is needed: the triggering only depends on the associated triggering probability.

2.4.3 *Dealing with multiple obstacles and side effects per task*

It is possible to have several obstacles and several side effects within the same task. In this case, three additional rules govern which obstacles and side effects can be triggered:

- At most one obstacle can be triggered during a task execution.
- Any number of side effects can be triggered during a task execution, up to the author-defined limit specified for a given interactive narrative.
- An obstacle can be triggered along with any number of side effects, up to an author-defined boundary. For each side effect, the author has to specify: a) with which obstacle it can be triggered (if any); b) whether this obstacle-related triggering is exclusive or not. If the obstacle-related triggering is exclusive, the side effect will only trigger when the associated obstacle triggers.

2.4.4 *Obstacle triggering with multiple relations*

In the general case, an obstacle can be related to several obstructions, clearing, counter-goaling, and subgoaling relations. According to the above rules, some of these relations would set the obstacle on, while other would set the obstacle off. We propose the following algorithm to resolve such conflicts:

- If there are only subgoaling relations, the obstacle is off if and only if all subgoals are reached.
- Otherwise, select the most recent out of all counter-goaling, obstruction, and clearing relations of the conflict set and use that relation to determine whether the obstacle is on or off.

This algorithm prioritizes counter-goaling, obstructions, and clearing over subgoaling.

2.5 Graphical notations

Given our narrative-oriented key concern, we consider it essential to support graphical representations for all relations, as illustrated in Figure 1 and Figure 2.

Figure 1 introduces the graphical notation for PS-101 first-class elements while Figure 2 depicts the relations between elements.

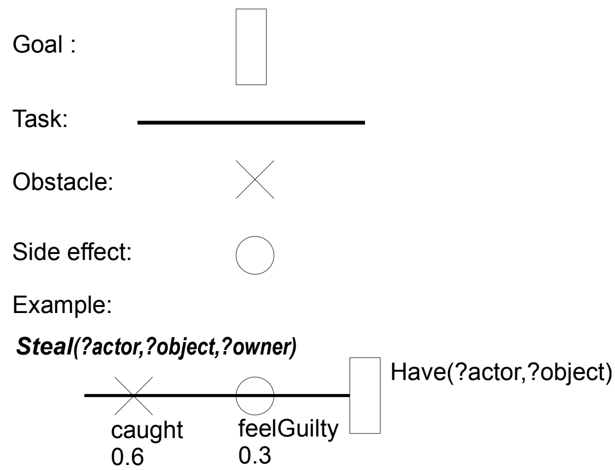

Figure 1: Graphical notation of the first-class elements of the PS-101 model of performative structure. The example at the bottom denotes a goal *Have(?actor, ?object)* with a single task *Steal(?actor, ?object, ?owner)* associated to it, to which one obstacle (*caught*, risk 0.6) and one side effect (*feelGuilty*, triggering probability 0.3) are associated. The task shares its first two parameters (?actor and ?object) with the goal and includes an additional third parameter (?owner).

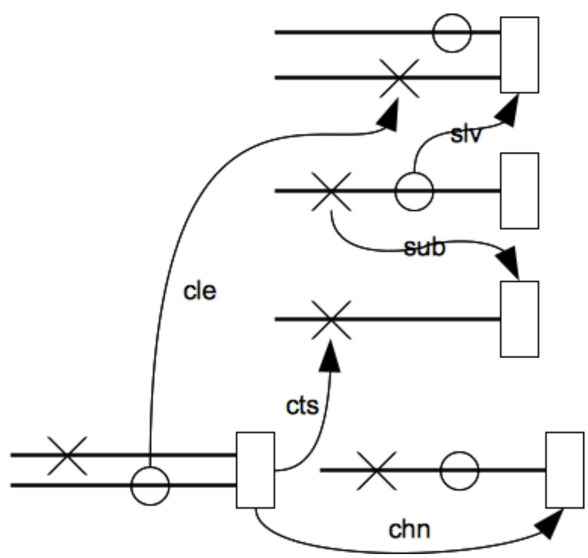

Figure 2: A PS-101 performative structure representing various relations between its elements. These relations are graphically represented by arrows, with a label indicating the type of relation and the matching conditions (if any).

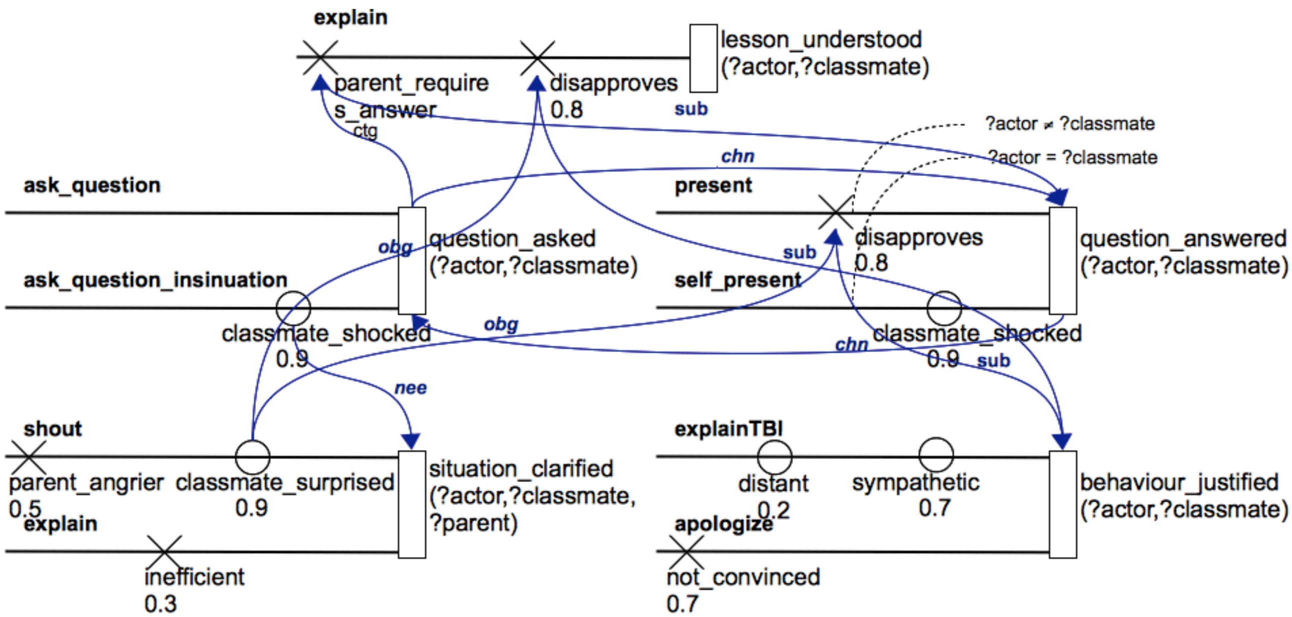

Figure 3: Example of a PS-101 performative structure. Goals, tasks, obstacles, side effects and relations are depicted as explained in Figure 1. Where task parameters are not specified explicitly, they are inherited from their goal parameters. Similarly, obstacle parameters inherit parameters from their task. The dashed lines attached to tasks represent the specification of task conditions (see Section 2.2.2). Matching conditions associated to relations are not depicted here for the sake of readability. Entities and initial conditions are also not shown.

3. AN EXAMPLE

To illustrate the PS-101 model, we give a concrete example based on an on-going research project investigating the use of interactive narratives for psychological assistance [8]. The story features Frank, a teenager who has to deal with his father, Paul, who is suffering from a Traumatic Brain Injury (TBI). Paul often shows inappropriate behaviours that are difficult to deal with. In the particular situation of the example, the family is at home and Julia, Frank's classmate, comes to visit him to ask him some question regarding the last math course. While Frank, played by the player, wants to explain the lesson to his friend, his father is constantly asking questions about Julia. Frank needs to answer them, otherwise Paul will get aggressive, making it impossible to focus on explaining the math. Some of Paul's questions are inappropriate insinuations about the relation between Frank and Julia, and Frank needs to clarify the situation. Frank can react somewhat aggressively to his father, which is quite understandable if the family context is known, but Julia is not aware of all this, and she is sometimes unclear about her friend's reactions. Frank then has to justify his behaviour. The graphical PS-101 rendering of this story is illustrated in Figure 3.

The performative structure depicted in Figure 3 must be completed with the set of entities: Frank (the teenager), Paul (his father), Julia (Frank's classmate), Lili (Frank's sister), Grandma (Frank and Paul's mother) and the initial conditions:

- lesson_understood(Frank,Julia),
- question_asked(Paul,Julia).

In this scenario, several elements are repeated during the unfolding of the story. In particular, there is an oscillation between the goals *question_asked* and *question_answered*. This is typical of Interactive Storytelling, in which limited material is meant to produce many situations. However, the feeling of repetition can be attenuated rather easily, e.g. by the natural language generation component, for which several variants of the same task, obstacle, or side effect can be added at this level. In this way, narrative actions are repeated, but not their concrete physical realization.

4. ON EVALUATION

Based on the design features of PS-101, we propose the following criteria for evaluation:

- *Ease of use*: are users (authors, story engineers) comfortable and efficient with an authoring tool based on PS-101?
- *Clarity:* are mappings of stories to PS-101 structures (cf. the example in the previous section) readily understandable to users other than the original authors and do the interpretations of users agree?
- *Expressivity*: Is PS-101 representation judged to be close to what the users wanted to express?
- *Quality of produced story*: Are the stories produced with PS-101 "better" (e.g. more complex; or using more nonlinear features) than stories produced with existing modeling approaches?

Building an experimental setup for such an evaluation is challenging, because it requires having two (versions of) systems, one with PS-101 and one without it, training of a sufficient number of authors on them, and measuring the criteria on practical authoring exercises. In addition, exogenous influences such as general usability of the authoring tool and the actual use(s)/exploitation of PS-101 structures by narrative engines need to be isolated and understood. Existing research-based Interactive Storytelling systems have rarely been evaluated from the authoring point of view, especially because the training phase requires too much effort. Systems such as Façade, Storytron, IDtension, or Mimesis require a significant amount of time to master. Some form of evaluation is possible when the system comes with a fully functional authoring environment that enables, typically, classroom experimentation. This is the case of *Wide Ruled*, for which it was possible to analyze quantitatively the story produced by three classrooms [26].

Our current evaluation plan is based on working with our in-house interactive drama engine, IDtension, which implements both PS-101 features and more classical pre-condition /post-condition management of actions. Informally, we already observed that the PS-101 related narrative elements and relations appear to “speak” to the author. This “hybrid system” will be tested by a dozen of game design students. Ease of use, clarity, and expressivity will be evaluated for both classical and PS-101 structures.

In addition to such empirical evaluation, PS-101 can be assessed against the documented features of existing approaches and systems. In Façade, for example, the author has to learn a programming language to be able to enter content at the behavioral level [18]. Planning-based systems require the author to enter pre-condition and post-conditions for every operator, which, as we could observe, remains a difficult and counter-intuitive challenge. The Wide Ruled study referred to above confirms that complex Boolean logical conditions are not an intuitive concept. Character-centric approaches [1,17] require the handling of the cognitive architecture underlying each agent, which constitutes a difficult task. Such systems have motivated specific authoring approaches, which start from a desired action, and infer, automatically [25] or manually [16], underlying agents’ parameters.

5. CONCLUSION

In this paper, existing models for Interactive Narrative have been revisited in order to lay the foundation of a different related approach. Instead of focusing on algorithms and computational constraints, we added the two perspectives of narrative relevance and authorability, applied to just a subpart of a whole interactive narrative system: the performative structure. The proposed model, PS-101, comprises four basic elements and nine relations between these elements, all of which can easily be depicted graphically.

Alongside its evaluation, the documentation material of PS-101 is being improved, and candidates for extension of the range of modelling primitives are being collected. These extensions will likely include relations with temporal conditions (such as timeouts) and hierarchical organization of narrative elements.

6. REFERENCES

[1] Aylett, R. 1999. Narrative in virtual environments – towards emergent narrative. In *Proceedings of the AAAI 1999 Fall Symposium on Narrative Intelligence,* Technical Report FS-01-02, AAAI, Menlo Park, 83-86.

[2] Barber, H. and Kudenko, D. 2007. Dynamic generation of dilemma-based interactive narratives. In *Proc. Third Conf. on Artificial Intelligence and Interactive Digital Entertainment – AIIDE,* AAAI Press, Menlo Park, CA, 2-7.

[3] Bates, J. 1994. The role of emotion in believable agents. *Communications of the ACM* 37(7):122-125.

[4] Brauer, W. and Reisig, W. 2009. Carl Adam Petri and ”Petri nets”. *Fundamental Concepts in Computer Science* 3:129–139.

[5] Bremond, C. 1973. *Logique du récit*. Seuil, Paris.

[6] Cavazza, M., Charles, F. and Mead, S. J. 2001. Characters in search of an author: AI-based virtual storytelling. In *Proceedings of the First International Conference on Virtual Storytelling – ICVS*, LNCS 2197, Springer Verlag, 145-154.

[7] Cavazza, M., Charles, F. and Mead, S. J. 2003. Generation of humorous situations in cartoons through plan-based formalisations. In *Proc. of the ACM CHI 2003 Workshop on Humor Modeling in the Interface*.

[8] Dumas, J. E., Szilas, N., Richle, U., and Boggini, T. 2010. Interactive simulations to help teenagers cope when a parent has a traumatic brain injury. *Computers in Entertainment - Special Issue: Advances in Computer Entertainment Technology*. 8, 2 (Dec. 2010)

[9] Egri, L. 1946. *The art of dramatic writing*. Simon & Schuster, New York London Toronto Sydney Tokyo Singapore.

[10] Endraß, B., Boegler, M., Bee, N. and André, E. 2009. What Would you do in their shoes? Experiencing different perspectives in an interactive drama for multiple users. In *Proc. of the Int’l Conf. on Interactive Digital Storytelling – ICIDS*, LNCS 5915, SpringerVerlag, 258-268.

[11] Field S. 1984. *Screenplay – The Foundations of Screenwriting*. 3rd edition. Dell Publishing, New York.

[12] Finin, T., Fritzson, R., McKay, D., McEntire R. 1994. KQML as an agent communication language, in Adam N.R., et al.(eds.), *Proc. 3rd Int’l Conf. on Information and Knowledge Management (CIKM94)*, ACM, New York, 456-463.

[13] Greimas, A. J. 1966. *Sémantique structurale*. Presses Universitaires de France, Paris.

[14] Jung B., Petta P.: Improving upon the TAEMS/DTC framework in the context of coordinated scheduling. in Trappl R. (ed.): *Proceedings of the Seventeenth European Meeting on Cybernetics and System Research* (EMCSR 2004), Austrian Society for Cybernetic Studies, Vienna.

[15] Klesen, M., Szatkowski, J., and Lehmann, N. 2000. The Black Sheep – Interactive improvisation in a 3D virtual world. In *Proc. of the i3 Annual Conference*.

[16] Kriegel, M., Aylett, R., Dias, J., Paiva, A. 2007. An authoring tool for an emergent narrative storytelling system. In: B.S. Magerko et al. (eds.) Intelligent Narrative Technologies. Papers from the AAAI Fall Symposium, Technical Report FS-07-05, AAAI Press, Menlo Park, CA, 55-62.

[17] Machado, I., Martinho, C. and Paiva, A. 1999. Once upon a time. In *Proceedings of the AAAI 1999 Fall Symposium on Narrative Intelligence*, Technical Report FS-01-02, AAAI, Menlo Park, 115-119.

[18] Mateas, M., and Stern, A. 2000. Towards integrating plot and character for interactive drama. In *Working notes of the Social Intelligent Agents: The Human in the Loop Symposium*. AAAI Fall Symposium Series, AAAI Press, Menlo Park.

[19] Meehan, J. TALE-SPIN. In: R. Schank and C. Reisbeck, (eds.) *Inside computer understanding: Five programs plus miniatures*. Lawrence Erlbaum Associates, Hillsdale, NJ, 197-226.

[20] Parunak, H. V. D. 1996. Visualizing agent conversations: using enhanced Dooley graphs for agent design and analysis.

In *Proc. of the 2nd Int'l Conf. on Multi-Agent Systems – ICMAS*, AAAI Press, Menlo Park CA, 275-282.

[21] Rank, S. 2005. Toward Reusable Roleplayers Using an Appraisal-based Architecture. *Applied Artificial Intelligence* 19(3-4), 313-340. DOI=10.1080/08839510590910138.

[22] Riedl, M., Saretto C.J., Young, R.M. 2003. Managing interaction between users and agents in a multi-agent storytelling environment. In. AAMAS '03, ACM, New York, 741-748

[23] Sacerdoti, E. D. 1975. *A structure for plans and behavior*. Technical Note 109, SRI Artificial Intelligence Center.

[24] Sgouros, N. M. 1999. Dynamic generation, management and resolution of interactive plots. *Artificial Intelligence* 107(1), 29-62.

[25] Si, M., Marsella, S.C. and Pynadath, D.V. 2005. Thespian: Using multi-agent fitting to craft interactive drama. In *Proc. International Conference on Autonomous Agents and Multi Agent Systems*, ACM Press, New York, 21-28.

[26] Skorupski, J. and Mateas, M. 2009. Interactive story generation for writers: Lessons learned from the Wide Ruled authoring tool. In *Proc. of the 8th Digital Art and Culture Conference (DAC 2009)*, Univ. of California, Irvine.

[27] Spierling, U. and Szilas, N. 2009. Authoring issues beyond tools. In *Proc. Int'l Conf. on Interactive Digital Storytelling – ICIDS*,,LNCS 5915, Springer Verlag, 50-61.

[28] Swartjes, I., and Theune, M. 2006. A Fabula model for emergent narrative. In *Proc. of Technologies for Interactive Digital Storytelling and Entertainment – TIDSE 2006*, LNCS 4326, Springer Verlag, 49-60.

[29] Szilas, N. 1999. Interactive drama on computer: Beyond linear narrative. In *Proceedings of the AAAI 1999 Fall Symposium on Narrative Intelligence*, Technical Report FS-01-02, AAAI, Menlo Park, 150-156.

[30] Szilas, N. 2003. IDtension: a narrative engine for interactive drama. In *Proc. 1st Int'l Conf. on Technologies for Interactive Digital Storytelling and Entertainment – TIDSE*. Frauenhofer IRB Verlag, 187-203.

[31] Szilas, N. 2007. A Computational Model of an Intelligent Narrator for Interactive Narratives. *Applied Artificial Intelligence* 21(8), 753–801. DOI=10.1080/08839510701526574.

[32] Szilas, N. 2010. Requirements for computational models of interactive narrative. In Finlayson M. (ed.) *Proc. AAAI 2010 Fall Symposium on Computational Models of Narrative*, 62-68.

[33] Szilas, N., Marty, O, and Rety, J.-H. 2003. Authoring highly generative interactive drama. In *Proc. 2nd Int'l Conf. on Virtual Storytelling – ICVS*, LNCS 2897, Springer Verlag, 37-46.

[34] Thue, D., Bulitko, V., Spetch, M. and Wasylishen, E 2007. Interactive storytelling: A player modelling approach. In *Proc. 3rd Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE*), AAAI Press, 43-48.

[35] Todorov, T. 1969. *Grammaire du Decameron*. Mouton, The Hague – Paris.

[36] Vale, E. 1973. *The technique of screenplay writing*. 3rd edn. Universal Library Edition.

[37] Wagner, T., Horling, B., Lesser, V., Phelps, J. and Guralnik, V. 2003. The struggle for reuse: Pros and cons of generalization in TÆMS and its impact on technology transition. In : Satayadas, A., Dascalu, S.M. (eds.) : *Proc. 12th Int'l Conf. on Intelligent and Adaptive Systems and Software Engineering – IASSE*, 118-123.

[38] Young, M. R. 1999. Notes on the use of plan structure in the creation of interactive plot. In *Proceedings of the AAAI 1999 Fall Symposium on Narrative Intelligence*, Technical Report FS-01-02, AAAI, Menlo Park, 164-167.

[39] Young, M. R., Riedl, M. O., Branly, M., Jhala, A., Martin, R. J. and Saretto, C. J. 2004. An architecture for integrating plan-based behavior generation with interactive game environments. *Journal of Game Development* 1(1):51-70.