Story Manager in 'Europe 2045' Uses Petri Nets

Cyril Brom[1], Vít Šisler[2], and Tomáš Holan[1]

[1] Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic
{brom,holan}@ksvi.mff.cuni.cz
[2] Charles University, Faculty of Philosophy, Prague, Czech Republic
vsisler@gmail.com

Abstract. *Europe 2045* is an on-line multi-player strategy game aimed at education of high-school students in economics, politics, and media studies. The essential feature of the game is that players face various simulated scenarios and crises addressing contemporary key issues of the unified Europe. These scenarios are branching and can evolve in a parallel manner. In this paper, we present a technique for specifying plots of these scenarios, which underpins the story manager of *Europe 2045*. The technique is based on a modification of Petri Nets. We also detail one particular scenario concerning the current crisis in Darfur. On a general level this paper discusses the strengths and weaknesses of implementation of Petri Nets in virtual storytelling.

1 Introduction

The idea of using computer games to support training and learning objectives is more than 30 years old [4]. Recent works have explored the potentialities of commercial strategy games and simulations in formal education and their alleged advantages over classical e-learning and edutainment tools, e.g. [6]. Indeed, many of such games have been experimentally integrated to formal curricula in the last four years. Perhaps the most prominent case studies have been conducted with *The Sims 2*, *Civilization III*, and *Europe Universalis II* [6, 12, 15], but other attempts exists as well. The results from these pilots are promising, but also ambiguous in some aspects, e.g. [15]. Hence, so called "serious" or "educational" games are starting to achieve increasing amount of attention. These games are, contrary to commercial games, *intentionally developed as educational tools*, which makes their integration into formal education easier. For example, a role-playing game prototype *Global Conflicts: Palestine* has been recently evaluated in a Denmark high-school with positive outcome [7]. Another studies are being conducted, including *FearNot!*, an anti-bullying educational game [1], and *Revolution*, a multi-player educational role-playing game concerning American War of Independence [5, 8].

As a part of European funded project "Integration of IT Tools into Education of Humanities" we develop an educational game *Europe 2045*, which is likely the first on-line multi-player strategy game worldwide aimed at education of high-school students in economics, politics, and media studies. The implementation part is finished and the game is presently being tested. Five preliminary studies have been already carried out, each with about 10 high-school or undergraduate university

M. Cavazza and S. Donikian (Eds.): ICVS 2007, LNCS 4871, pp. 38 – 50, 2007.

students. A large-scale pilot evaluation is planed for November 2007 in a high-school in Prague, Czech Republic. The game is intended to be fully applied in spring 2008.

Europe 2045 features three layers of game-play. Each student (1) represents one EU member state in the game and is responsible for its governmental policies, economical development, and social issues. Additionally, in cooperation with the other players, (2) he or she is engaged in setting politics of the whole EU. Nevertheless, the essential feature of the game is that (3) each player faces various simulated scenarios and crises addressing contemporary key issues of the unified Europe, including migration, population aging, international relations, and energy independence. Not only have these scenarios a strong educational potential, but also they introduce storytelling into the game. Storytelling has played an important role in humanities education since the advent of formal schooling ([4]). Stories help to build a learning context, through them the students can better understand the problematic, they increase their involvement, and consequently their motivation.

Specifying plots of stories and controlling the course of a game in accordance with these plots is a well known problem. It was indeed one of the most challenging goals we faced during the development. Essentially, the game had to be designed in order to meet the following requirements:

a) the story plots to be branching,
b) the story episodes to be both global, i.e. concerning the whole Europe, and local, i.e. concerning a particular state or a set of states,
c) the episodes to can happen in parallel, because we have more than 20 countries, which could be played simultaneously, each having defined different episodes,
d) the episodes to be triggered by various initial conditions depending on the time, EU economy etc.,
e) the technique for specification of the plots to be intuitive enough for a high school teacher or another user (typically an undergraduate university student of humanities) to be able to design new scenarios for the game.

Finally, we have chosen a modification of Petri Nets [2] as the plot specification technique. Although this modification is used for *Europe 2045*, the technique is quite universal and can be use in other applications as well. The goal of this paper is to present this technique and discuss its strengths and weaknesses. We first detail the game *Europe 2045* in Section 2, focusing on how the stories are narrated in the game. In Section 3, we review previous research on methods of controlling a story in games and storytelling applications in general. Section 4 details the Petri Nets modification and its implementation. Section 5 demonstrates a part of a scenario concerning the Darfur crisis. Section 6 concludes.

2 'Europe 2045'

This section details the game *Europe 2045*. We first describe the game from the perspective of a player, focusing on its storytelling aspects, and then overview the technical background.

Europe 2045 is an on-line multiplayer game in which each student (i.e. a player) governs one state by setting its policies, taxes, and subsidies while discussing

European and global issues with other players. The game contains economical and social model which simulates population aging, migration, evolution of the market, transfers of industry and services, changes in environment, moods of citizens, and a substantial number of other variables describing particular states and European Union as a whole (e.g. culture, infrastructure, education, etc.).

On the European level, all the players are encouraged by the game to take active part in decision making. The narrative structure of *Europe 2045* serves for three purposes. First, it introduces new topics and agenda for students' discussions. Second, unfolding new events in accordance with players' previous decisions, it serves as a global feedback for the students and as a method for sharpening the discussion. Both these kinds of events are global, i.e. they are common for all the players and concern EU as well as international issues (e.g. conflict in Darfur has intensified). The third class of events provides individual players with a feedback about the results of their previous actions concerning their own states; hence, these events are local (e.g. citizens in France protest against university fees, or unemployment in Czech Republic has reached 15%).

The game proceeds in rounds, one round is one game year. An atomic "beat" of a scenario is called an *affair*. It is an event that takes place in one round and can be triggered by players' actions or results from the economical and social model or affairs from previous rounds. An affair is communicated to the player via a textual description in the game newspaper (*NP news item*) or via a short animation in TV, which is being displayed at the beginning of every round (*TV news item*). In some cases, an affair also has an *impact* on the economical and social model, i.e. it influences state of a country or the whole EU. Typically, an affair can result in increasing the EU budget, increasing the level of pollution in particular states, crippling agriculture production, etc.

Some affairs introduce issues that require decision to be taken by the players (e.g. accepting another state's proposal, sending humanitarian mission to the area of a conflict, etc.). These decisions are intended to be taken during a discussion, typically in the class under the teacher's supervision, and voted through a *ballot*. One affair often triggers more ballots, each constituting precisely formulated question ("Do you vote for sending European humanitarian mission to Darfur area?") with three possible answers (yes/no/abstain). The ballots chosen by the game designers aim to cover all the main possible solutions usually proposed by real politics in similar cases. When the answers can not be schematized to the yes or no option, the ballot contains number (3-4) of more detailed solutions. The decision chosen by the players influences the economical and social model and the affairs to be triggered in the next round.

The game offers more different campaigns to be played, each of them focusing on different problematic (e.g. energy independence, international relations, environment). For each campaign, specific affairs and a scenario describing relations between them have to be designed. New campaign also comprises distinctive animations for the TV, articles for the newspaper, items for the in-game encyclopaedia, teachers manual, and handouts for students.

Technically, the game is a client-server application; the students play the game via the Internet (Fig. 1). The server part comprises PHP scripts generating the game interface, the story manager written in PHP as well, and the social-economical simulation, which is written in Java. Almost all parts of the interface are programmed in Flash (see Fig. 2). The social-economical simulation features a simplified model of EU economy. Technically, the model is a multi-agent simulation [16], where each agent is either a country, or an abstract representation of an EU industry, like travel industry, mining industry, agriculture etc. In a simplified fashion, at the end of each round, an agent–country computes next state of the country, while an agent–industry carries out decisions in which country to build new factories, mines etc. based on particular variables of the countries (e.g. mining industry agent would prefer countries with low environmental tax, travel industry agent would prefer countries with nice environment, high culture and developed infrastructure). We remark that this simulation is coarse grained in the sense that it does not feature human-like agents.

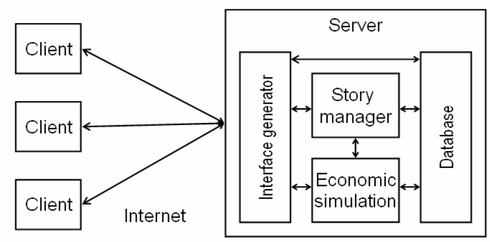
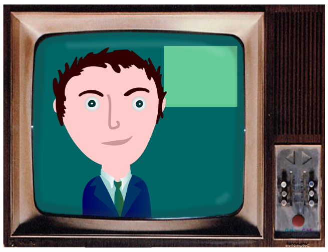
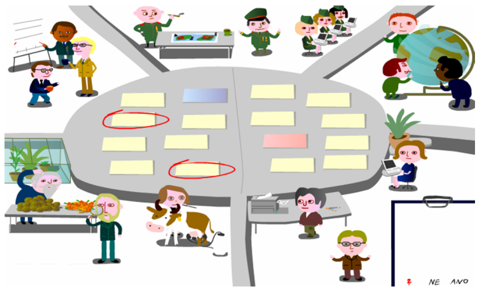


Fig. 1. Architecture of *Europe 2045*



2a

2b



2c

Fig. 2. a) The TV news. b) The interface, through which the player governs its country. c) The balloting interface.

3 Related Work

The issue of generating/controlling stories in games and storytelling applications is notoriously known. Most techniques come from games and experimental simulations featuring human-like actors. As said above, *Europe 2045* does not employ such actors. Rather, the story events are more abstract; they deal with whole populations, with a country economy etc. However, formally, the problem is very similar. We remind that we had several requirements on the storytelling technique in *Europe 2045* ranging from parallelism of the stories to the technique to be of use to undergraduate university non-IT students (see Sec. 1).

A well known branch of techniques for specifying plots are deterministic finite-state machines (dFSMs) [e.g. 13, 14]. Each state represents a story episode, and a transition is a trigger that detects the end of the episode and starts a next one. Natural advantage of dFSMs is that they are formal, and yet graphical (Fig. 3), which makes them easily intelligible. However, a classical dFSMs was not suitable for us, since they cannot cope with the issues of parallelism (req. b), c)). On the other hand, non-deterministic FSMs can cope with it, but they are not easily comprehensible (e)). Similarly to dFSMs, we encountered these parallelism difficulties when considering adopting the "beat approach" by Mateas [9].

In the field of emergent narrative, planning formalism is often used [1, 3, 11]. This technique can cope well with the requirements a), b), c), d), but it is not too friendly for a non-AI expert (req. e)). Since we are interested in pre-specified plots but not in automatic story construction (because the story must fit into the formal curricula), to use the HTN ("hierarchical-task network") formalism would be like using a sledgehammer to crack a walnut, with having the unintelligibility disadvantage. To tackle e), one could introduce a "presentation layer" for an HTN system to disguise the underlying representation and develop an authoring interface; however, this is time-consuming activity.

We needed something that would have *natural* comprehensibility advantage like the dFSMs, but could cope with the parallelism at the same time. The best candidate technique we found was Petri Nets, which is a specification technique frequently used in software engineering. Petri Nets have been already employed in storytelling. Natkin & Vega [10] used them to a retrospective analysis of a computer game story. In our previous work [2], we used it to prototype a story plot of a large simulation featuring human-like agents. However, none of these work implemented a story manager for a real full-fledged game, which is the case of *Europe 2045*.

Actually, many variants of Petri Nets exist. For our purposes, we specified our own modification, which will be detailed in the next section.

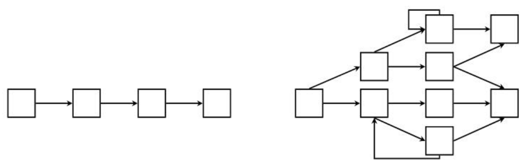


Fig. 3. Story plots as dFSMs. The linear plot is on the left, the branching on the right.

4 Petri Nets in 'Europe 2045'

This section gives description of our modification of Petri Nets. Generally, Petri Nets consist of containers (or places, represented by a circle: ○), tokens ("the pellets": ●), actions (or transitions, ▭), and transition function (⟶). The containers contain the tokens. If a sufficient number of tokens is contained in specific containers, an action is triggered. After firing an action, the tokens that helped to fire this action are removed, and some new tokens are generated (see Fig. 4a). Which tokens fire which action and which action generates tokens to which containers is specified by the transition function (and depicted by arrows). At one instant, several containers can contain tokens, which allows for concurrent triggering of actions (as opposed to dFSMs). The types of containers, tokens, and transition functions vary a modification from modification. For more thorough introduction to Petri Nets, we recommend the reader to consult [10].

The important feature of Petri Nets is that they are formal, and yet allow for graphical depiction, which mirrors all of their (or most of their, depending on the complexity of a particular Petri Nets variant) formal features. In other words, Petri Nets are a formal system, which has its own presentation layer. Hence, they fit well as a specification interface between a designer and a programmer. Indeed, in *Europe 2045*, we use our Petri Nets modification in two ways. First, they are employed in an informal manner as a specification tool for a game designer (and for university students who develop new game episodes as a part of their course). Second, a rigorous counterpart of this informal specification tool presents the architectural underpinnings of the story manager developed by our programmer. Thanks to the advantages of Petri Nets, the conversion of specifications into code is straightforward.

Our modification of Petri Nets was demanded to mirror the features of stories of *Europe 2045* described in Sec. 2. Particularly, the model must have seized the round-based nature of the game, the affairs, the ballots, and the presentation of news items. Hence, the model works with two types of actions, one of which is further coupled with the so-called ballot, and triggering of actions happens only between two rounds. These two features are also the most significant distinction of our model from a typical Petri Nets. We now describe individual components of our model. Then, we describe how they are integrated together and introduce the story manager algorithm.

Actions. We have two types of actions: *affairs* (▭), and *news items* (⬚). Both of them can be started between two rounds by a trigger, as described later. When an affair is triggered, it can influence the game by its game *impact*. An impact can be for example: "migration to EU decreases in this round by *x*", or "the EU budget increases in this round by *y* & agriculture production in the states *X*, *Y*, *Z* is crippled by the factor of *z*". This impact is immediate, i.e. its result is computed by the economic simulator before the next round starts (see Fig. 1). Contrary to affairs, news items never have a game impact. In a plot specification, impacts are not described graphically, only textually.

Both kinds of actions can communicate to the player several news items in the next round, using either the game newspaper, or the game TV. Only an affair can invoke (in the next round) one or more *ballots* about a proposal related to the affair.

When the next round finishes, both affairs and news items can generate a new token to a specific container. Fig. 4a depicts an action, which generates one token. In Fig. 4b, there is depicted an affair "Darfur conflict" invoking a ballot about "sending an EU mission to Darfur" proposal in the next round (which in case of agreement generates two tokens, each to a particular container – see next).

Ballots. In a typical ballot (⬭), each student has three possibilities: he or she can agree, disagree or abstain, but more complicated cases are also allowed. The result of a ballot can influence the game, however, not necessarily straight-forwardly. To determine, how the game is influenced (i.e. which game impact to apply), a *what-next function* is defined for each ballot. This function returns its value based on the voting result. The simplest what-next function is trivial: it returns "+" if the proposal was agreed and "–" if it was not. This function is used in Fig. 4b. A more complicated what-next function can for example return *A* if the total size of armies of the states that agreed is more than 75% of the size of all EU states' armies, *B* if it is between 50% and 75%, and *C* if it is less than 50%.

A game *impact* is then defined for each return value of a what-next function. Additionally, every ballot can generate one or more tokens similarly to actions. How many tokens and to which containers are generated again depends on the result of the what-next function. In a portrayal, the return values of the what-next function are depicted next to the arrows originating from the ballot and pointing to the containers to which the tokens are to be generated. What-next functions are specified textually.

Tokens. We employ state-less aging tokens, in contrast to our previous work [1], where we used aging tokens with state (in so-called coloured Petri Nets modification). A token starts to age after it is generated to a container, but not removed in the next round. The *age* is given in number of rounds the token stays in the container.

Triggers and Containers. Triggers are evaluated at the beginning of each round. We use three types of them. First, we have *token-generating triggers*. These are associated with containers. Similarly to [1], containers with the triggers are depicted as a double circle (◎), as opposed to containers without a trigger (○). Every time the condition of such a trigger holds, a new token is generated into the container. Presently, we use only the form of "when the round *x* starts → generate one token". These triggers essentially start the game or schedule the episodes to particular rounds. The number of a round, i.e. *x*, is depicted next to the container, as in Fig. 2b.

Second type of triggers is an *action-starting trigger*, whose meaning is obvious. Theoretically, an action-starting trigger cannot start its action directly for a conflict between two triggers can rise because they want to remove the same token (Fig. 4c). Hence, if the condition of a trigger holds, the trigger only marks its action to be started, and the respective tokens to be removed.

A typical action-starting trigger has the following form: "if there are *n* tokens in a particular container & *f* → mark the desired action", where *f* is a function of game state returning a boolean value (e.g. *f* can be "is EU budget in this round bigger then *x* EUR?"). The condition can also question the age of a token. In the portrayal, we write the ID of the trigger condition next to the arrow pointing to the action, as depicted in Fig. 4d. The condition itself is specified textually. Nothing next to the arrow, or only the age, means the condition: "if there is at least 1 token (of the specified age)".

To cope with conflicts, we define *conflict resolving triggers* that unmark one of the marked actions. This solution is based on our previous work [2]. The default conflict-resolving mechanism is to unmark one action randomly, but more elaborated solutions can be used, e.g. prioritising the actions. However, note that these triggers have been introduced to have the mechanism theoretically consistent; so far, we have not used them in our scenarios for they are not too intelligible for non-IT experts.

To sum up; in our Petri Nets modification, we use tokens without states but with aging, we use two types of actions (news items and affairs), and we can couple an affair with one or more ballots. Instead of transition function, we use three types of triggers, which can have relatively complicated conditions. Tokens can be generated by both ballots and actions.

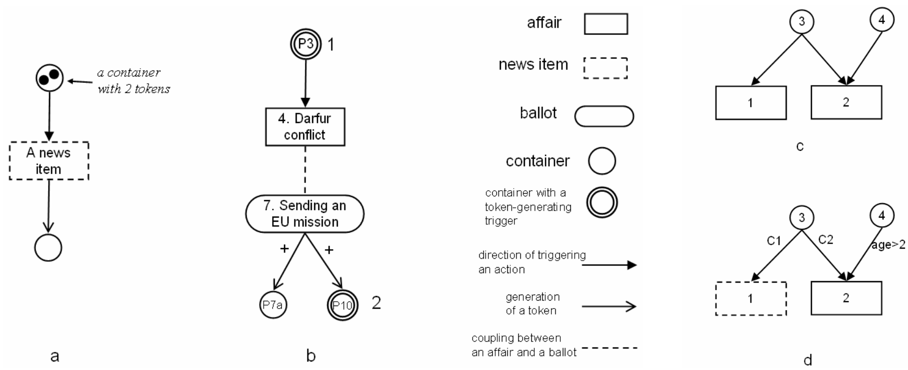


Fig. 4. a) A simple net. One token is needed in the upper container to start the news item, which then generates a token to the bottom container. b) A token-generating trigger generates a token to Container P3 in the first round. In fact, this starts the scenario. Affair 4 invokes Ballot 7 immediately. Between round 1 and 2 and if the what-next function of Ballot 7 returns "+" (which is in the simplest case when the ballot proposal has been agreed), two tokens are generated; one to P7a, one to P10. Additionally, a token-generating trigger generates one token into P10 in the second round regardless of the result of the ballot. c) If there is both one token in Container 3 and one token in 4, it is not clear whether to start Action 1 or 2. A conflict resolving trigger must be invoked. d) News item 1 is started when Condition C1 holds. Affair 2 is started when condition C2 holds *and* there is at least one token in Container 4 older than 2 rounds. Note, that in fact, the overall condition of trigger starting Affair 2 is "C2 & age_of_token_in_Cont4>2". If there is a conflict between this trigger and the trigger starting News Item 1, the conflict-resolving trigger is invoked as in the case (c).

1) Evaluate all token-generating triggers and generate tokens based on the results.
2) Evaluate all action-starting triggers and mark the respective actions to be started and the tokens to be removed.
3) For every token that is marked more than once: trigger the appropriate conflict-resolving trigger and consequently unmark one or more actions.
4a) If there is no marked action, then **end** this scenario.
4b) Otherwise, prepare all the marked actions to be run: generate newspaper & TV news based on the news items, consider the impact of the affairs, prepare the ballots of the affairs.
5) Remove all marked tokens, increase the age of remaining tokens.
6a) Run new round. Display the news, and ballots, ...
6b) End the round.
7) Calculate the results of the ballots & their what-next functions.
8) Generate new tokens based on the run actions and on the what-next functions of the ballots.
9) Compute the game impacts of the ballots based in their what-next function.
10) Go to 1

Fig. 5. The story manager algorithm. Note, that the steps 1-5 are concerned with a round k, while 7-10 with the round $k+1$.

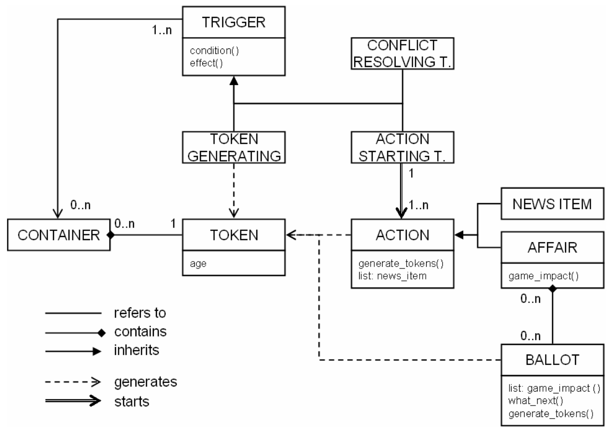


Fig. 6. Overall description of Petri Nets we uses given in a UML-like class diagram. Based on this schema, the story manager in Europe 2045 works. Relations of the "Conflict resolving trigger" are not depicted for clarity.

Game designer specifies the plots according to the above mentioned principles. Additionally, each entity of the model has a unique ID, which serves as a referencing mechanism to additional textual description (concerning e.g. the trigger conditions) as well as corresponding animations and news articles.

Petri Nets serves also as an architectural ground for our story manager. During a game, the plot specification is evaluated between each two consecutive rounds using the algorithm in Fig. 5. We remark that theoretically, some difficulties may be encountered in marking of tokens in Step 2 (for each trigger, we need to mark the tokens of appropriate age that has not been already marked, which may require searching for an appropriate ordering of triggers). Additionally, ordering of actions in Step 4b may matter (the total impact of "action A after action B" can differ from the total impact of "action B after action A"). Thought these issues are of theoretical interest, we did not tackled them rigorously for they could be addressed easily in an *ad hoc* manner even for our largest scenarios (Fig. 7).

Fig. 6 overviews how all the components of the model are integrated together. In the next section, we give an example of a large plot called "Darfur conflict".

5 Darfur Example

One of the largest and perhaps the most informative plots is *Darfur scenario* (Fig. 7). For brevity, we will not describe here the whole plot, rather we will illustrate how the building blocks of our Petri Nets work on the plot.

This scenario starts in the first round by an affair communicating via TV that the crisis in Darfur has escalated and invoking four ballot proposals. Based on the results of the ballots, the crises further develop. The important point is that it can evolve in several branches at the same time. For example, if students agree both on a form of development aid (Ballot 8) and a humanitarian aid (Ballot 5), both the affairs "Development aid begins" (8a) as well as "Humanitarian aid begins" (5a) are triggered in the second round. Additionally, either Affair 10a "Big migration increase", or Affair 10b "Small migration increase" is started. Which affair is started depends on conditions 7A and 7B.

The trigger conditions, ballot results and what-next-functions are not depicted graphically. They are rather specified textually and referred via IDs from the portrayal. Hence, the trigger starting Action 10a "Big migration increase" has the condition "if there are less than 3 tokens in P10" (Condition 7A), while the trigger starting 10b wants "at least 3 tokens" (Condition 7B). Notice also, that Container P10 has a token-generating trigger, which generates one token into P10 in the second round. This means that even if all the proposals are disagreed, large increase in migration to EU still occurs.

Similarly, it is specified textually that in the ballot "Lost reaction" (12), students have three possibilities: to reinforce the mission, to pull out the mission or to ask NATO for help. The what-next-function of this ballot is trivial (this is again specified textually) – to which container will be generated a token depends only on the result of the ballot (ids: 12A, 12B, 12C). It is also specified that Ballot 12 has no game impact, but the affair "Mission failed" (20) has game impact: "migration to EU is increased by *x*".

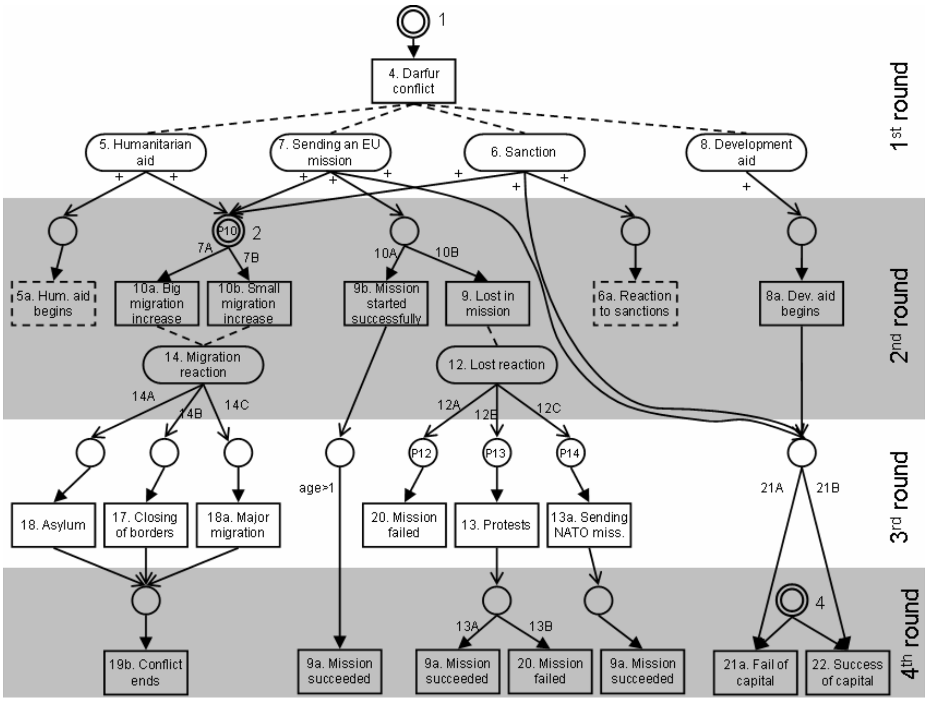


Fig. 7. Darfur scenario. The elements of the plot are organised according rounds for clarity. Container's ids that are not referred from text are not depicted. For clarity, several actions are depicted twice (9a, 20). Compare this figure with Fig. 3. We think that such scenario would be hard to describe by a deterministic FSM.

The scenario ends in the fourth round by a final series of affairs. However, notice, that other scenarios can be unfolded in parallel. In the basic campaign of *Europe 2045*, there are two additional scenarios of the size of “Darfur”, and several dozens of small scenarios comprising from one to three actions. The campaign lasts 10 rounds (game years).

6 Discussion and Conclusion

In this paper, we have introduced serious game *Europe 2045*, which is a multi-player round-based strategy game aimed at education of high-school students in economics, politics, and media studies. The game is fully implemented and it is presently being evaluated. A large-scale pilot is planed for November 2007 in a high-school in Prague, Czech Republic. The game is intended to be applied in spring 2008.

A player in the game faces, among other things, various simulated scenarios addressing contemporary key issues of the unified Europe. This essentially introduces storytelling into the game. We have addressed the issues of specifying plots of the scenarios, and of unfolding stories by developing a Petri Nets modification that serves both for the purposes of a game designer and as an architectural ground for the story manager in the game.

The strength of Petri Nets is that they (at the same time) allow for graphical and yet formal description, they are easily comprehensible, and allow for describing branching stories evolving in parallel. The comprehensibility was demonstrated by the fact that the technique was explained to college students of humanities during a course (during about an hour and half) and they were subsequently able to use it to specify their own “toy” campaigns. For this strength, we favoured Petri Nets over deterministic finite state machines, beat-approach of Michael Mateas, and HTN planning formalism. Additional advantage, which we did not use however, is that Petri Nets can be run independently on the underlying simulation, as demonstrated in [2]. This helps with testing plots (e.g. are all the parts of the story reachable?).

Nevertheless, it must be noticed that Petri Nets fit well only for stories that are preset, not for emergent narrative. We also think that the beat approach is better for controlling stories featuring relatively small virtual worlds inhabited by (a few) virtual humans – this approach is more flexible. Additionally, if one needs a story with preset plot, which however do not evolve in parallel, deterministic FSM would be likely sufficient. Finally, it must be stressed that our technique is a branching one: after all, the author must specify all branches in advance. The potential risk of combinatorial explosion of branches must be avoided by manual “cutting” by the author.

We have developed one 10-round campaign for *Europe 2045*, which comprises about 70 game events (affairs, or news items). The plot of this campaign is programmed directly in PHP. This allowed us a quick start, but it also presents a limitation. To facilitate the development process, we would benefit from a graphical authoring tool, especially because we aimed at creating a second campaign and several undergraduate humanities students, who do not know PHP, develop other campaigns as a part of their university course. Developing this tool presents our future work. However, note that even in this tool, Petri Nets would be presented to the user *as such*, there would be no need for adding another presentation layer disguising the underlying representation (as may be the case of HTN planning – see Sec. 3). In fact, Petri Nets are a sort of rule based system, and it is this underlying rule based system that they innately present in a graphical form, further, Petri Nets can be even viewed as a methodology constraining the space of possible rules that can be written down.

Acknowledgments. The research on usage of Petri Nets in storytelling was partially supported by the Program "Information Society" under project 1ET100300517, and by the Ministry of Education of the Czech Republic (Res. Project MSM0021620838). The project "Integration of IT Tools into Education of Humanities" is financed by the European Social Fund, the state budget of the Czech Republic, and by the budget of Municipal House Prague. The authors would like to thank to all the partners of the project: Generation Europe, Ciant, gymnasium Sázavská, and Association for International Affairs, and to all the people who helped, most notably to Petr, Edita, Jakub, Lenka, Ondřej, Martin, and Michal.

References

1. Aylett, R.S., Louchart, S., Dias, J., Paiva, A., Vala, M.: FearNot! – An Experiment in Emergent Narrative. In: Panayiotopoulos, T., Gratch, J., Aylett, R., Ballin, D., Olivier, P., Rist, T. (eds.) IVA 2005. LNCS (LNAI), vol. 3661, pp. 305–316. Springer, Heidelberg (2005)
2. Brom, C., Abonyi, A.: Petri-Nets for Game Plot. In: Proceedings of AISB Artificial Intelligence and Simulation Behaviour Convention, Bristol, vol. 3, pp. 6–13 (2006)
3. Cavazza, M., Charles, F., Mead, S.J.: Planning Characters' Behaviour in Interactive Storytelling. The Journal of Visualization and Computer Animation 13, 121–131 (2002)
4. de Freitas S.: Learning in Immersive worlds: A review of game-based learning. JISC (Joint informational Systems Committee) report (2006) (June 6, 2007), http:// www.jisc. ac.uk/ eli_outcomes.html
5. The Education Arcade: Revolution, a role-playing game (June 6, 2007), http://www.educationarcade.org/revolution
6. Egenfeldt-Nielsen, S.: Beyond Edutainment: Exploring the Educational Potential of Computer Games. PhD Thesis, University of Copehagen (2005)
7. Egenfeldt-Nielsen, S., Buch, T.: The learning effect of 'Global Conflicts: Middle East'. In: Santorineos, M., Dimitriadi, N. (eds.) Gaming Realities: A Challenge for Digital Culture, pp. 93–97. Fournos, Athens (2006)
8. Francis, R.: Revolution: Student's experiences of virtual role play within a virtual reconstruction of 18th century colonial Williamsburg (an unpublished manuscript)
9. Mateas, M.: Interactive Drama, Art and Artificial Intelligence. Ph.D. Dissertation. Department of Computer Science, Carnegie Mellon University (2002)
10. Natkin, S., Vega, L.: Petri Net Modelling for the Analysis of the Ordering of Actions in Computer Games. In: Proceedings of Game-ON, pp. 82–92 (2003)
11. Reidl, M.O., Stern, A.: Believable agents and Intelligent Story Adaptation for Interactive Storytelling. In: Göbel, S., Malkewitz, R., Iurgel, I. (eds.) TIDSE 2006. LNCS, vol. 4326, pp. 1–12. Springer, Heidelberg (2006)
12. Sandford, R., Ulicsak, M., Facer, K., Rudd, T.: Teaching with Games. Using commercial off-the-shelf computer games in formal education, Futurelab, Bristol, UK (June 6, 2007), www.futurelab.org.uk/download/pdfs/research/TWG_report.pdf

13. Sheldon, L.: Character Development and Storytelling, ch. 7, 14. Thompson Course Technology (2004)
14. Silva, A., Raimundo, G., Paiva, A.: Tell Me That Bit Again.. Bringing Interactivity to a Virtual Storyteller. In: Balet, O., Subsol, G., Torguet, P. (eds.) ICVS 2003. LNCS, vol. 2897, pp. 146–155. Springer, Heidelberg (2003)
15. Squire, K.: Replaying history: Learning World History through playing Civilization III. PhD thesis, Indiana University (2004)
16. Wooldridge, M.: An Introduction to MultiAgent Systems. John Wiley & Sons, Chichester (2002)