Scenario analysis based on Linear Logic

Frédéric Collé
frederic.colle@univ-lr.fr

Karim Sehaba
karim.sehaba@univ-lr.fr

Michel Augeraud
michel.augeraud@univ-lr.fr

L3I - Université de La Rochelle
Av M. Crépeau
F-17042 LA Rochelle

ABSTRACT

This paper deals with game analysis. We begin by the proposal of a terminology relevant to game scenario. Then, we propose an approach dedicated to analyze game scenarii. This approach consists in expressing a scenario by using a fragment of linear logic. This model is then translated into a Petri net model. This model allows us to generate the possible narratives for the given scenario that are at last analyzed. Linear logic facilitates the creation of the model: in the case of a complex scenario, the Petri net is difficult to build. However, we use the Petri net as an operational model.
Morever, in addition to usual provable properties such as liveness and safety properties, we propose a new class of properties establishing conclusions with respect to the scenario relevance.

Keywords

narratives, game analysis, linear logic

1. INTRODUCTION

This paper aims to present how to use Linear Logic to model and validate concurrent interactions within a strategy multi-player game. Indeed, [17] underlines the miss of analysis techniques for games. Our proposal is twofold: first a terminology of game is given and then a new checking method of a model expressed in Linear Logic is proposed.

This paper is dedicated to the translation of a model written in linear logic into a Petri net in order to generate accessible sequences of actions for a given scenario.

In addition to usual provable properties such as liveness and safety properties, we propose a new class of properties establishing conclusions with respect to the scenario relevance.

After a short introduction of analysis problems within the scope of strategy games, we define a class of properties which can be checked in such a system. Then, we present a linear logic based method for modeling game scenarii. At last we propose an analysis methodology of this model as well as solutions obtained from an example.

2. SOME GAME-SPECIFIC PROBLEMS

2.1 Existing approaches

The game domain has been subject to particular attention recently, not only in the computer science domain but also in creation and design.

Several categories of video games arose according to either historical, editorial or narrative criteria. Usual distinguished categories of games are action, adventure, strategy and sport. [16] proposes an other classification based on game mechanisms. Firstly he distinguishes "one-player" games, where the player plays alone against the computer, and "multi-player" games. He also talks about cooperative games, where the players play together against the computer, and competitive games, where the players play, alone or with a team, one against the other. Each case is then divided into several classes: puzzle, strategy game, action game and adventure game.

We are interested, within the scope of this paper, in strategy games in a multi-player frame. A "one-player" strategy game gives the player the feeling of controlling his universe. A "multi-player" strategy game is a game in which each player can influence the universe, but but actions of other players have an influence on the player's universe too.

Video games editors tend to create more and more scenario-based games to make them more attractive (players can be the game's heroes and follow a pre-defined plot line).

Several teams deal with such a problem. We can quote the "Liquid Narrative Group [27, 26, 25, 20], the "Zero game studio" [5, 10], the Salford university team [2, 13, 11, 12], the Michigan university team [14], the Teesside university group [4] and also the OZ project [15]. All those studies concern the difficulty in generating a coherent game execution (compared to game rules and game scenario) according to the player's actions.

Indeed, in order to make a game be attractive for the player, it is necessary that the latter has the feeling of controlling

the world in which he evolves and of being able to modify it in his own way. However to put a scenario in a game tries to bring back the player to the same-and-unique plot line. Generally authors aim to write a coherent game unfolding. This construction is done during the game's execution either with a previously made analysis (such a choice was made by the "Zero game studio" team which uses causal graphs) or by adding constraints which must remain valid during the game's execution (like in Teesside university works).

All of those works are generally based on Artificial Intelligence techniques. So, the property concept is used but no property is never expressed. The only team, as far we know, working on game analysis is the CNAM's one [17, 24]. Their work concerns the analysis and the specification of video games. They propose an analysis technique allowing to guarantee that actions carried out by the player remain coherent within the game. They use Petri net based construction techniques allowing to guarantee that when each sub-network is coherent, the set built by respecting a set of rules is coherent.

2.2 Our selected terminology

Authors treating on games agree on the fact that the terms which are associated to narration description in a game are not fixed and are prone to be debated. We don't aim to debate on this subject, however we will define the terms and the concepts used in this paper:

- *The action concept* is such as a stage in dramatic art. Although this concept is never defined, it is very significant insofar as it is often a question of a course of actions or a course of events.
- *The event term*, closer to the computer science community, is defined as follows:

  Definition 1. *An* event *is the atomic element of a story, a significant thing which occurs.*

  We underline that an event can occur from the history, generated by the designer (e.g. appearance of a dragon when the player enters a localization) or from the user (e.g. when the player opens a door).
- According to this definition of the event notion, we can immediately define a *narrative.* A narrative is often regarded as a story. But the notion of story is generally specified. For example [12] considers the theoretical story as being what the author wishes to see, in opposition to the story (which is held). [14] consider story as all the events which appear and the constraints between those events. That can also be seen like the storyline.

  So, we can consider that the narrative is included in something which represents the set of possible stories (the narration for [21], the story world for [20]). In order to tackle this confusion between the terms of narrative and story, we define the notion of narrative by:

  Definition 2. *A* narrative *is an ordered sequence of events.*

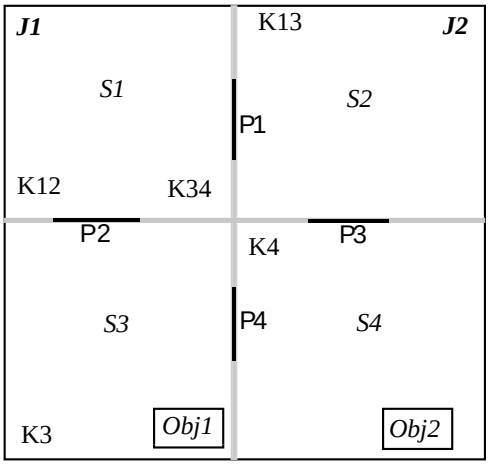


Figure 1: Topological description of our example

  whereas we define the set of the possible stories as a *scenario* in the following way:

  Definition 3. *A* scenario *is the set of ordered sequences of events.*

- At last, we add the notions of *person* and *resource*:

  Definition 4. *A* person *is an entity which takes part in the scenario events.*

  Definition 5. *A* resource *is an element which can be used by a person.*

2.3 Presentation of our strategy example

In order to illustrate our proposal, we use an example of an exploration game. It is a multi-player game. The players are located in a space made up of rooms which are connected each to others by doors. The objective of each player is to move across those rooms to find an object. The game is finished when a player found his object. In order to move from a localization to the another, the player must go through a door. He must possess the key dedicated to the door. Each key opens a particular door or a set of doors. The keys are of single use, and the door is closed again after the player has gone through. A key thus doesn't allow to go more than once through a door.

The figure 1 presents the topology of our example. The four doors (ressources) P1, P2, P3 and P4 are dedicated respectively to cross from S1 to S2, S1 to S3, S2 to S4 and S3 to S4. Each door can be crossed in a bi-directional way.

Five key resources are arranged in the world: k12 is located in S1 and allows to open the doors P1 and P2, k34 is located in S1 and allows to open the doors P3 and P4, k13 is located in S2 and allows to open the doors P1 and P3, k3 is located in S3 and allows to open the door P3 and k4 is located in S4 and allows to open the door P4.

We remark that in this example the users are totally free to modify the world. The narrative aspects are not present

but could appear for example as an introduction of new keys according to relevant users actions.

At last, this game reveals four categories of events: the introduction of a player in the game, the acquisition of a key by a player, the door crossing done by a player and the achievement of the player objective (to take the object).

A narrative will be then for example: ("introduction of player 1 "; "introduction of player 2"; "player 1 takes the K12 key"; "player 1 goes through the P2 door"; "player 1 takes the Obj1 object).

A scenario will be described by an initial state, the set of the events which can occur during the game and a desired final state. Such a model allows us to produce a set of narratives which constitutes the story.

2.4 Classification of game-specific properties

We define here the classes of properties which can be expressed and checked within the scope of the validation of game scenarii before their execution.

In order to build the set of system properties to be validated, we use the usual classes of properties such as safety or liveness properties. However, the checking of a scenario requires to take into account of an other set of properties directly related to the relevance of the scenario. Indeed, we must guarantee that the game course will not conduce the player in an erroneous state or a blocking state ("does the game correctly finish ? ", "Are the players able to win?", "Does it exist an unreachable room?"...) and we also have to establish some conclusion on the quality of the scenario ("Do the players have equal chances to reach their goal ?", "Is the game not "too easy "?...).

It is then possible to distinguish two main classes of properties: One which is related to the *playability* of the scenario (Is the game course correct?), the other allows to evaluate *the relevance of the scenario* (i.e is the scenario interesting?). We describe here these two property classes.

2.4.1 Playability

The checking of the playability of the game is very strongly connected with the usual checking of a distributed system. We verify properties of safety, liveness, reachability or absence of deadlock. We recall the definition of these properties [1].

- **Safety property**: express that under certain conditions, an event can never occur.
- **Liveness property**: express that under certain conditions an event will end up taking place.
- **Reachability property**: indicate that a state of the system can be reached.
- **No deadlock property**: express that the system will be never in a situation where it cannot progress.
- **Fairness property**: express that, under certain conditions events will infinitely often arrive (or will not arrive).

2.4.2 Checkings relevant to the scenario pertinence

The originality of our approach is due to the definition of an additional class of properties relevant to the game field. Indeed, to define a playable scenario is not enough to ensure its validity. A game can be considered to be useful (or valid) if it is ludic, sufficiently difficult so that the player is not bored, and if the scenario does not promote one player over the others.

- **Impartiality** (or justice): the players have equal chances (or almost) to win the game.
- **Complexity** : the game course does not contain executions which lead to a too much fast victory.
- **Competition**: the interest of the game consists in putting the players in competition. For such a property we will make sure that there exists a sufficient number of situations where the players are in direct competition to take an object. It is this type of situation which can possibly make the game ludic.

These properties (in particular the scenario relevance) are relative properties. Indeed, it is necessary to determine before the checking phase which is the minimal number of executions conducing to the victory. Moreover, it is almost impossible to obtain a scenario for which a total impartiality is reached. However, it is necessary to try to obtain for each player a success rate close to 50 percent.

The objective announced here is not only to validate a given scenario but to determine the best scenario, i.e the scenario which satisfies all the properties.

3. A MODELING PHASE OF A SCENARIO IN LINEAR LOGIC

3.1 A quick overview of Linear Logic

Linear logic was proposed by J.Y. Girard[6] in 1987 like a refinement of first-order logic in order to take into account the concept of consuming and producing resources. Some structural rules were denied : the contraction rule and the weakening one. One of the consequences is that, in linear logic, two specimens of a formula are distinguished from only one specimen of this formula. So, this logic is well adapted in order to model systems involving resources.

The logical connectives are re-examined in this resource-based interpretation. Linear logic has many restrictions and variants dependently to the subset of the connectives used. The fragment of linear logic we chose to use is the multiplicative-additive one (MALL).Let us introduce the connectives used :

- *linear implication* is defined by the following rule :

$$\frac{\Gamma \vdash F \quad \Delta, G \vdash H}{\Gamma, \Delta, F \multimap G \vdash H} \multimap L$$

  Using linear implication allows to define how the system state changes. The proposition $F \multimap G$can be interpreted as :consume resource F to achieve resource B.

- *multiplicative conjunction* denotes simultaneous occurrence of resources. This connective is defined by :

$$\frac{\Gamma, \varphi, \psi \ \vdash \ \Delta}{\Gamma, \varphi \otimes \psi \ \vdash \ \Delta} \otimes L$$

- *additive disjunction* represents a choice over which one has no control. The rule is :

$$\frac{\Gamma \ \vdash \ \varphi, \Delta}{\Gamma, \vdash \ \varphi \oplus \psi, \Delta} \oplus R1 \quad \text{and} \quad \frac{\Gamma \ \vdash \ \psi, \Delta}{\Gamma, \vdash \ \varphi \oplus \psi, \Delta} \oplus R2$$

Those connectives allow us to define logic propositions in order to reason about game scenarii. Reasoning on linear logic proposition is made by a sequent calculus. A sequent calculus consists in the re-write of the sequent by replacing the connectives by their introduction rules in order to obtain the meta-connector " , " in the formulae. This produces the canonical proof tree. A sequent is valid if all the sheets of the proof tree correspond to identity rules.

We will write a sequent by placing in the antecedent of the sequent both the initial state of the game and the desired events and in the succedent of the sequent the awaited final state.

3.2 A model for our example

A model of our toy-example consists of six categories of atoms and four categories of events:

Atoms the initial state of the player i is represented by the atom J_i. The player i in the room j is defined by J_{iSj}. The atom k_x represents the key allowing to open the set of the doors x. k_{xJ_i} models the key k_x possessed by the player i. Obj_i represents the objective of the player i and J_{iF} the end of the game for the player i.

Events The event corresponding to the introduction of a player in the game is done by the formula $J_i \multimap J_{iS_j}$. The event corresponding to the acquisition of the key allowing to open the set of the doors x by the player i is $J_{iS_j} \otimes k_x \multimap J_{iS_j} \otimes k_{xJ_i}$. The event corresponding to the crossing of the player i from the room j towards the room z by going through one of the doors of the set x using the possessed key k_x is given by $J_{iS_j} \otimes k_{xJ_i} \multimap J_{iS_z}$. At last, the event corresponding to the achievement of the objective of the player i is given by the formula $J_{iS_j} \otimes Obj_i \multimap J_{iF}$.

Then, the scenario is done by the following sequent : $J, OBJ, K, E \ \vdash \ J_{iF}^{\oplus}$ where J is the set of the player atoms, OBJ the set of the objective atoms, K the set of the key atoms, E is the set of the events and $J_{iF}^{\oplus}$ the set of the final states.

The scenario represents the set of parameters which determine the possible executions of the game. In order to obtain a valid game, we present in the sequel of this paper the types of properties which have to be satisfied by the scenario.

4. SCENARIO ANALYSIS

In this section, we present the analysis which can be deduced from the model expressed in Linear Logic. This analysis is based on the study of the sequent calculus.

By translating the sequent in a Petri net we obtain the set of the narratives from the scenario. We can then deduce the scenario properties from this set.

4.1 An overview of our proposed methodology

Several approaches for the proof of a sequent of Linear Logic were proposed [22]. Among all these approaches we take as a starting point P. Kngas's one [8, 9]. The author proposed to translate a sequent of Linear Logic in a Petri net. Accordingly, the writing of a proof of a sequent corresponds to the firing of a sequence of transitions starting from an initial marking and driving to the awaited final marking.

Indeed, [7] demonstrated that the act of proving the validity of a sequent is equivalent to find a firable sequence of transitions in a Petri net. I.e, let R be a Petri net with its initial marking M_0 and a final marking M_f, and let $\overline{s}$ be a non-ordered list of transitions: there is an equivalence between the act of proving the sequent $M_0, \overline{s} \vdash M_f$ and the act of finding a sequence σ_j of transition firings conducing from M_0 to M_f in the network such as the transitions of σ are those of $\overline{s}$ with the same arity.

4.1.1 Translation in a Petri net

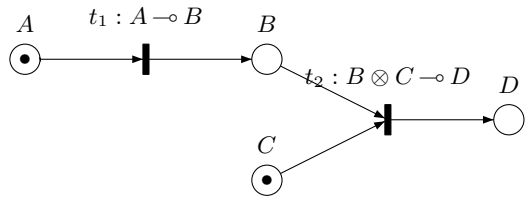
P. Küngas proposed a translation of a sequent in a Petri net. This construction implies the use of the "cut" rule in order to write a proof. However, [3] shows that the use of this rule forces to reason on the whole of the Petri net whereas a sequence doesn't involve all the elements of the network. So, we will retain the following translation, inspired of the works of the team "Rseaux de Petri et Logique" of the LAAS/CNRS [23, 19, 18]. The translation of the linear logic model in a Petri net model is made in the following way:

- For each atomic proposition, we associate a place of the Petri net. The availability of the proposition implies the presence of a token in this place.
- An implicative formula is translated by a transition.
- The antecedent of the sequent consists both of a set of implicative formulae and of propositional formulae. It corresponds at the same time to the set of all transitions to be fired and the initial marking. The propositional formulae in the succedent of the sequent correspond to the final marking.

For example, let the following model: $A, C, A \multimap B, B \otimes C \multimap D \ \vdash \ D$

The corresponding Petri net is shown in the following figure:

4.1.2 Generation of the proof tree



For the previous example, the sequent calculus conduces to the following proof tree:

$$\dfrac{\dfrac{}{A \vdash A}\,id \qquad \dfrac{\dfrac{\dfrac{}{B \multimap B}\,id \quad \dfrac{}{C \multimap C}\,id}{B, C \multimap B \otimes C}\,\otimes R \qquad \dfrac{}{D \vdash D}\,id}{B, C, B \otimes C \multimap D \vdash D}\,\multimap L}{A, C, A \multimap B, B \otimes C \multimap D \vdash D}\,\multimap L$$

This proof corresponds to the firing of the sequence (t_1;t_2) in the Petri net given in figure 4.1.1 whose the initial marking is a token in the places A et C.

A narrative is obtained in writing the proof of a sequent of Linear Logic. So, the previous proof shows that the scenario "consume A to achieve B", "consume B and C to achieve D" with the hypotheses "A" and "C" and the conclusion "D" can produce the narrative "consume A to achieve B" then "consume B and C to achieve D" (the proof is to be read in a bottom-up way).

We don't aim to prove the validity of a sequent, but we aim at obtaining all possible proof trees. So, the algorithm we use is the following:

- step 1 : sort the set of firable transitions $\overline{s}$
- while it exists an enabled transition of $\overline{s}$ which can be fired
  - For all transition s_i of $\overline{s}$
    - If s_i is enabled and $s_i \neq s_{i+1}$
      - fire s_i
      - remove s_i from $\overline{s}$
      - Return to step 1
    - else continue

By using this methodology, the Petri net model derived from the Linear Logic model presented in section 3.2 is shown in annex.

4.2 Properties to be satisfied

For our example, we can establish the following properties relevant to the playability of the game:

- **Safety:** A player can't go through a door without the dedicated key.
- **Reachability:** Each player can win the game.
- **Fairness:** A room can't be reached an infinity of time.
- **No deadlock:** The game can't reach a state where the players can't evolve.

The properties relevant to the scenario pertinence are:

- **Impartiality** (or justice): The two players have the same proportion of chance to reach their goal.
- **Complexity** : The players can't reach their goals without going through at least three doors.
- **Competition**: It exists some executions where the two players are in an immediate competition to take a key. Indeed, the dramatic intensity of the game can be measured by means of instances of direct competition between the players. Even if such a property is difficult to be checked with classical model-checkers, the analysis of execution traces gives quickly a result. A competition situation is for example a situation where the players are in the same room and where only one key which is useful to leave the room is available.

4.3 Checking results

The analysis of the traces enables us to establish some conclusions relevant to the properties concerning the relevance of the scenario. The analysis shows us that there is 22859 possible sequences for this scenario. A study of these traces allows us to provide the following results, relevant to the impartiality, complexity.

4.3.1 Analysis of the impartiality

The interest to use linear logic in order to analyse scenario pertinence is the simplicity of checking the impartiality property. Indeed, for obtaining the number of sequences where player 1 is victorious, one only has to determine the number of sequences ending in the transition $e43$. In the same way, we can count the number of sequences where player 2 is victorious (sequences ending in the transition $e42$).

Thus, there is 8220 sequences leading to the victory of J1 and 10176 sequences for the victory of J2 (we can note that there is 4463 executions for which no player wins). Among all the executions leading to the victory, player 2 wins in 55 % of the cases. So, we can considered that the scenario is equitable.

4.3.2 Complexity checking

In this paragraph, we aim to check the complexity of the scenario. So, we wish to ensure that it doesn't exist an execution where one of the players achieve its objective without going through at least three doors. One of the obtained sequences is the following:

```
['e42', 'e41', 'e1', 'e2', 'e9', 'e12', 'e13', 'e43']
```

This sequence corresponds to the execution where the player 1 takes the key $k34$, then moves towards the room 3 and achieves its objective. For this player, this execution leads to a too fast victory, it doesn't be thus of any interest.

5. CONCLUSION

In this paper we were interested in game analysis by using a model defined in linear logic. We have highlighted various relevant properties for a multi-player strategy game and we have proposed a new class of properties suitable for the checking of games, namely the class concerning the relevance of scenarii.

Then, we proposed a method allowing to evaluate the satisfaction of those properties for a given game scenario. This method consists to model a scenario by using linear logic. Linear logic is a suitable language to formally write games' scenarii. The interest, in using linear logic, is to focus in the expression of a game on actions and ressources without useless informations. This language is well suited to express locally actions and to check for possible orders. So, for games, it allows to easily check some properties concerning the relevance of scenarii, in particular the complexity and the impartiality. Then, a translation into Petri nets is used in order to generate possible narratives for the given scenario. Lastly, we analyze those narratives in order to evaluate if previous properties are or not satisfied.

One of the major benefits of our proposal is to help a scenario designer. So, this approach have to be integrated in a more general approach for constructing relevant game scenarii. Accordingly, one of the prospects for our approach is to use the validation results to choose the optimal scenario. Indeed, when we check relevance properties, results are evaluated according to an initial hypothesis. One of the possible methods should consist in generating all possible scenarii and in evaluating the satisfaction of relevance properties on these scenarii. The selected scenario should be then one of those scenarri which optimally satisfy relevance properties according to initial hypotheses.

Another prospect would consist in exploiting the structure of the Petri net in order to determine the liveness of the game as well as the absence of deadlocks. Lastly, a last prospect would be to take into account temporal constraints of the scenario. The suggested approach with linear logic would allow to reduce the combinatory explosion, because sequences generation takes account of the restriction of the marking due to the temporal reduction.

6. ADDITIONAL AUTHORS

Additional authors:
Armelle Prigent (L3I, `armelle.prigent@univ-lr.fr`)
and Ronan Champagnat (L3I, `ronan.champagnat@univ-lr.fr`).

7. REFERENCES

[1] A. Arnold. *Systmes de transitions finis et smantique des processus communicants*. Masson, 1992.

[2] R. Aylett and S. Louchart. Towards a narrative theory of virtual reality. *Virtual Reality, special issue on storytelling*, 7, 2004.

[3] R. Champagnat. Optimisation d'une squence de franchissement de transitions dans un rseau de petri t-temporis. In Y. Dallery, J. Hennet, and P. Lopez, editors, *MOSIM'O3 (MOdlisation et SIMulation)*, pages 94–100, Toulouse (France), Apr. 2003.

[4] F. Charles, M. Lozano, S. J. Mead, A. F. Bisquerra, and M. Cavazza. Planning formalisms and authoring in interactive storytelling. In *1st international conference on technologies for interactive di gital storytelling and entertainment*, Darmstadt (Germany), 2003.

[5] M. Eladhari. *Object oriented story construction in story driven computer gam es*. Master thesis, Department of History of Literature and History of Ideas Stockh olm University, Stockholm, 2002.

[6] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[7] F. Girault. *Formalisation en logique linaire du fonctionnement des rseaux de Petri*. Thesis, Universit de Toulouse III, Toulouse (France), Dec. 1997.

[8] J. Kngas. Using linear logic planning to make knowledge bases reactive. In *Seventh Symposium on Programming Languages and Software Tools*, pages 135–148, Szeged (Hungary), June 2001.

[9] P. Kngas. *Linear Logic programming for AI planning*. Master thesis, Tallinn technical university, Tallinn (Estonia), May 2002.

[10] C. Lindley. The gameplay gestalt, narrative, and interactive storytelling. In *Computer games and digital cultures*, Tampere (Finland), June 2002.

[11] S. Louchart and R. Aylett. Emergent narrative, requirements and high-level architecture. In *SETN*, 2004.

[12] S. Louchart and R. Aylett. The emergent narrative theoretical investigation. In *Narrative and learning environments conference NILE*, pages 24–33, Edinburgh (Scotland), 2004.

[13] S. Louchart and R. Aylett. Narrative theories and emergent interactive narrative. *IJCEELL*, 2004.

[14] B. Magerko and J. Laird. Building an interactive drama architecture. In *1st international conference on technologies for interactive di gital storytelling and entertainment*, Darmstadt (Germany), Mar. 2003.

[15] M. Mateas. *Interactive drama, art, and artificial intelligence*. Ph.d. thesis, Carnegie Mellon University (School of Computer Science), Pittsburg (USA), Dec. 2002.

[16] S. Natkin. *Jeux vidos et mdias du XXI sicle*. Ed. Vuibert, Paris, 2004.

[17] S. Natkin and L. Vega. A petri net model for the analysis of the ordering of actions in computer games. In *Proceedings of GAME'ON 2003*, 2003.

[18] B. Pradin-Chezalviel and R. Valette. Accessibilit de marquage et logique linaire dans un rseau de petri t-temporel. In *Journes Formalisation des Activits Concurrentes*, pages 123–134, Toulouse (France), May 2000. CERT-IRIT-LAAS/CNRS.

[19] B. Pradin-Chezalviel, R. Valette, and L. Knzle. Scenario durations characterization of t-timed petri nets using linear. In *PNPM'PP (Petri Nets Performance Modeling)*, Zaragoza (Espagne), Sept. 1999.

[20] M. Riedl and R. Young. Character-focused narrative planning. Submitted to Virtual Reality 2003, 2003.

[21] N. Szilas. Interactive drama on computer: beyond linear narrative. In *AAAI Fall symposium on narrative intelligence*, pages 150–156, 1999.

[22] T. Tammet. Proof strategies in linear logic. *Journal of Automated Reasoning*, 12:273–304, 1994. Also available as Programming Methodology Group Report 70, Chalmers University, 1993.

[23] R. Valette, B. Pradin-Chezalviel, and R. Champagnat. Raisonnement temporel pour la supervision et l'ordonnancement des systmes dynamiques vnements discrets. In C. ditions, editor, *le temps, l'espace et l'volutif cole thmatique document et volution du groupe de recherche information, interaction, intelligence du CNRS*, volume II, pages 103–117, 2000.

[24] L. Vga. *Un Modle bas sur les Rseaux de Petri pour l'analyse de l'ordonnancement des actions dans un jeu vido*. Thesis, CNAM, Paris (France), Nov. 2004.

[25] R. Young. Story and discourse: a bipartite model of narrative generation in virtual worlds. Submitted to Virtual Reality 2003, 2003.

[26] R. Young, M. Riedl, M. Branly, A. Jhala, R. Martin, and C. Saretto. An architecture for integrating plan-based behavior generation with interactive game environments. Submitted for Publication in Journal of Game Development, vanlent@icts.usc.edu and niles@charlesriver.com, 2003.

[27] R. M. Young. Steps towards a computational theory of interactive narrative in virtual worlds. Unpublished document, 2003.

APPENDIX
Annex

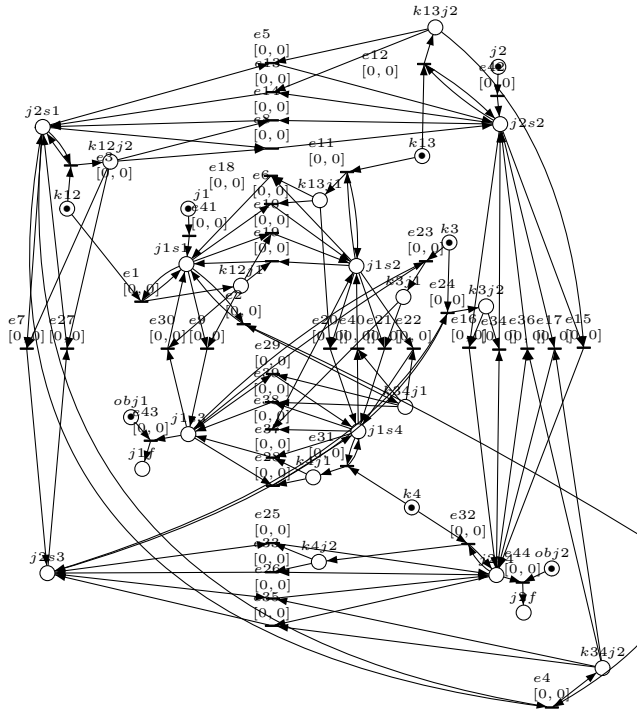


Figure 2: The Petri net model of our game