

# Study over story management in videogames

**Alessio Cecconi**  
**Joint work with Stavros Vassos**



**SAPIENZA**  
UNIVERSITÀ DI ROMA

## Goal of the project

- Study the state-of-the-art methods for designing stories in videogames
- Investigate how AI methods can be used to provide new tools for generating stories with some form of verification

# Interactive storytelling

- **Authored:** every possible unfolding of the story is manually written by writers/designers
  - **Linear:** unique predetermined storyline path
  - **Non-linear:** more plot alternatives, still fixed
- **Emergent:** the story is generated real-time in a goal-based manner, also reactive to user actions
  - Automated story unfolding according to “short-term” plot goals

## New approach

- **Verified:** emergent story with strong guarantees over all possible unfoldings
  - Instead of automatically choosing an unfolding that satisfies some “short-term” plot goal, take into account also “global” or “long-term” requirements, e.g., there is no future deadlock

# State of the art tools for story management

- We study the whole process of realizing a story
  - **Modelling** the story: specify the story conceptually, e.g., in normal English text
  - **Implementing** the story: formalize the story in a standard language or data structure
  - **Executing** the story inside a general-purpose game engine: use the formalized story to drive the gameplay
- Also we focus on any verifying capabilities of story unfoldings

# State of the art in authored stories

- Names of systems, ...

Tool	Modelling	Implementing	Execution	Verifying
Word processor				
Spreadsheet				
Graph builder				
Hypertext pages				
Story designer				
Visual scripting in game engine				
Dialog systems				
Interactive fiction engine				

# State of the art in authored stories

- No global standard for story representation
- Tools used to avoid direct coding
- Little or no verification capabilities

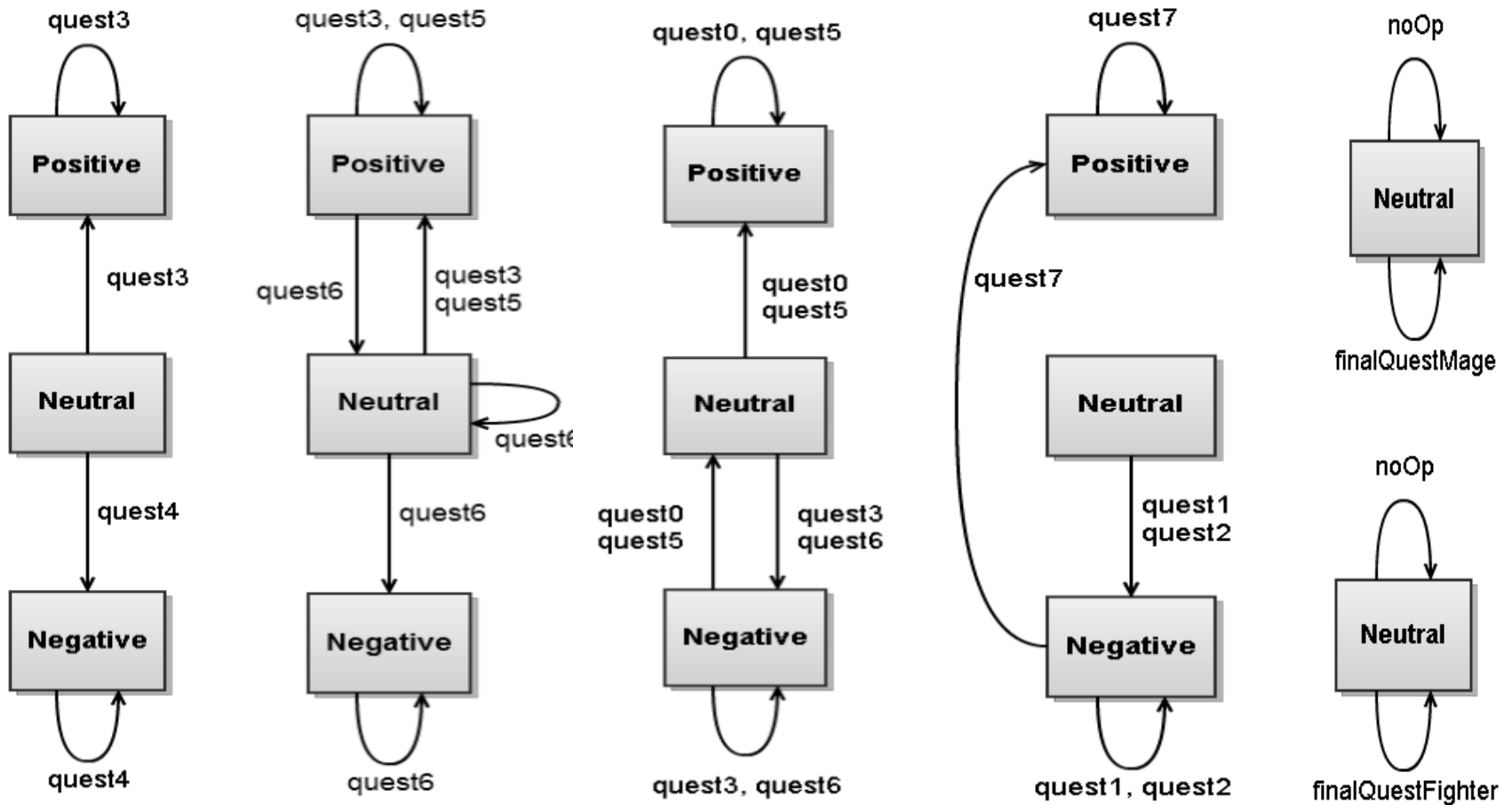
Tool	Modelling	Implementing	Execution	Verifying
Word processor	✓	✗	✗	✗
Spreadsheet	⚠	✓	✗	✗
Graph builder	✓	⚠	✗	✗
Hypertext pages	✓	✓	✗	✗
Story designer	✓	✓	✗	⚠
Visual scripting in game engine	⚠	✓	✓	✗
Dialog systems	✓	✓	⚠	⚠
Interactive fiction engine	✗	✓	✓	✗

## Behavior composition & JaCO framework

- Our approach about *verified stories* is based on **Behavior Composition**
- Realizing a virtual desired **target** (i.e. *story flow*) process expressed as a **transition system** by suitably coordinating the execution of a set of concrete available **behaviors** (i.e. *narration entities*), always as **transition systems**.
- Not a fixed sequence of events, but the result of independent entity's behavior orchestration

## Example. MageFighter (behaviors)

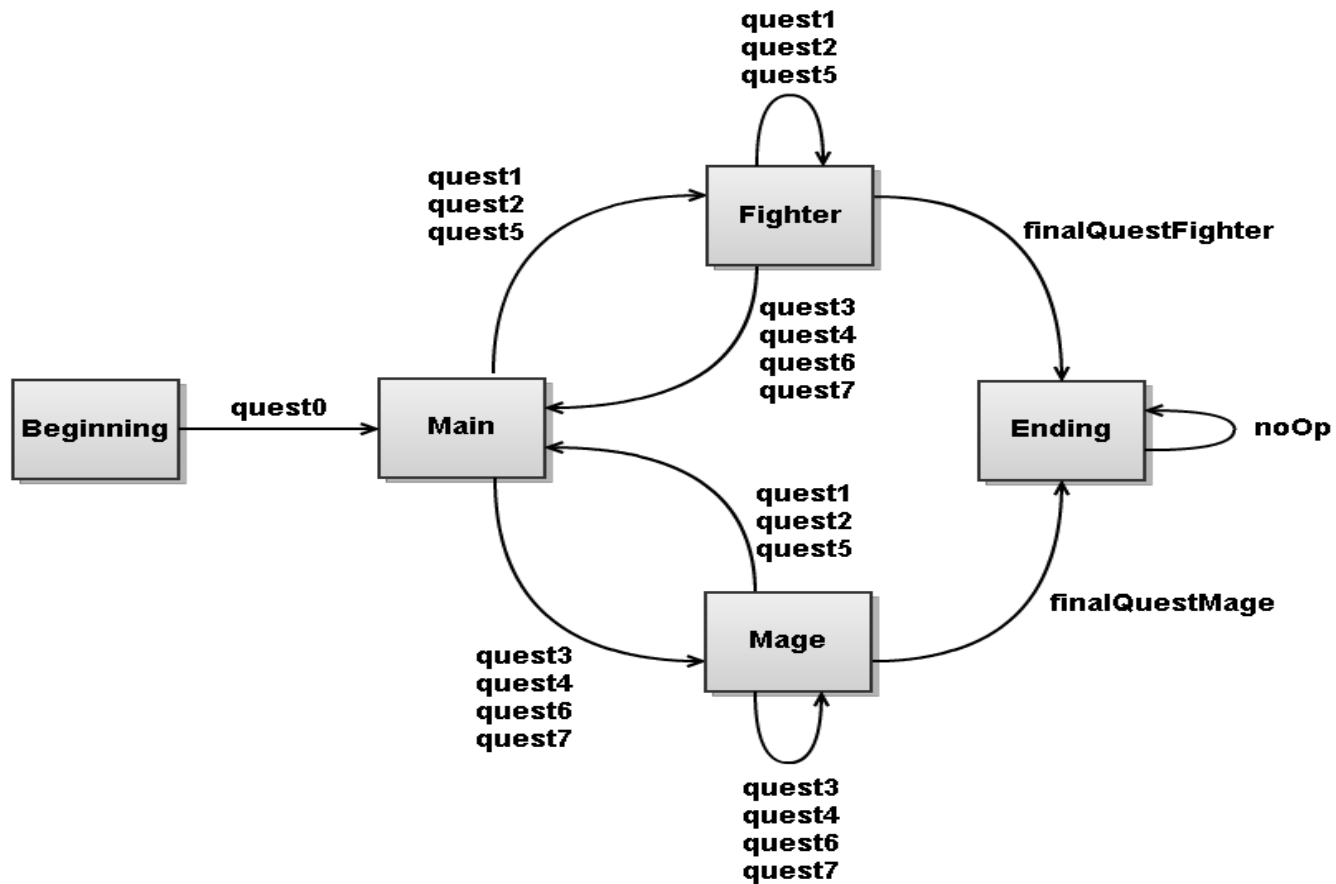
Many distinct behaviors...





## Example. MageFighter (target)

...compose a bigger (virtual) target.



## Behavior composition & JaCO framework

- **JaCO** (**J**ava-based **C**omposition-**O**riented) RESTful web service for behavior composition orchestration
  - **Input** : FSMs of *behaviors* and *target* , represented in XML
  - **Output** :
    - *Target relizable: Controller Generator (CG)*, a FSM lookup-table of any existing composition
    - *Target unrealizable*: Automatic identification of the unrealizable transition and configuration
- Certainty over story flow realizability in any possible behaviors configuration
- CG computed offline for an optimal online usage

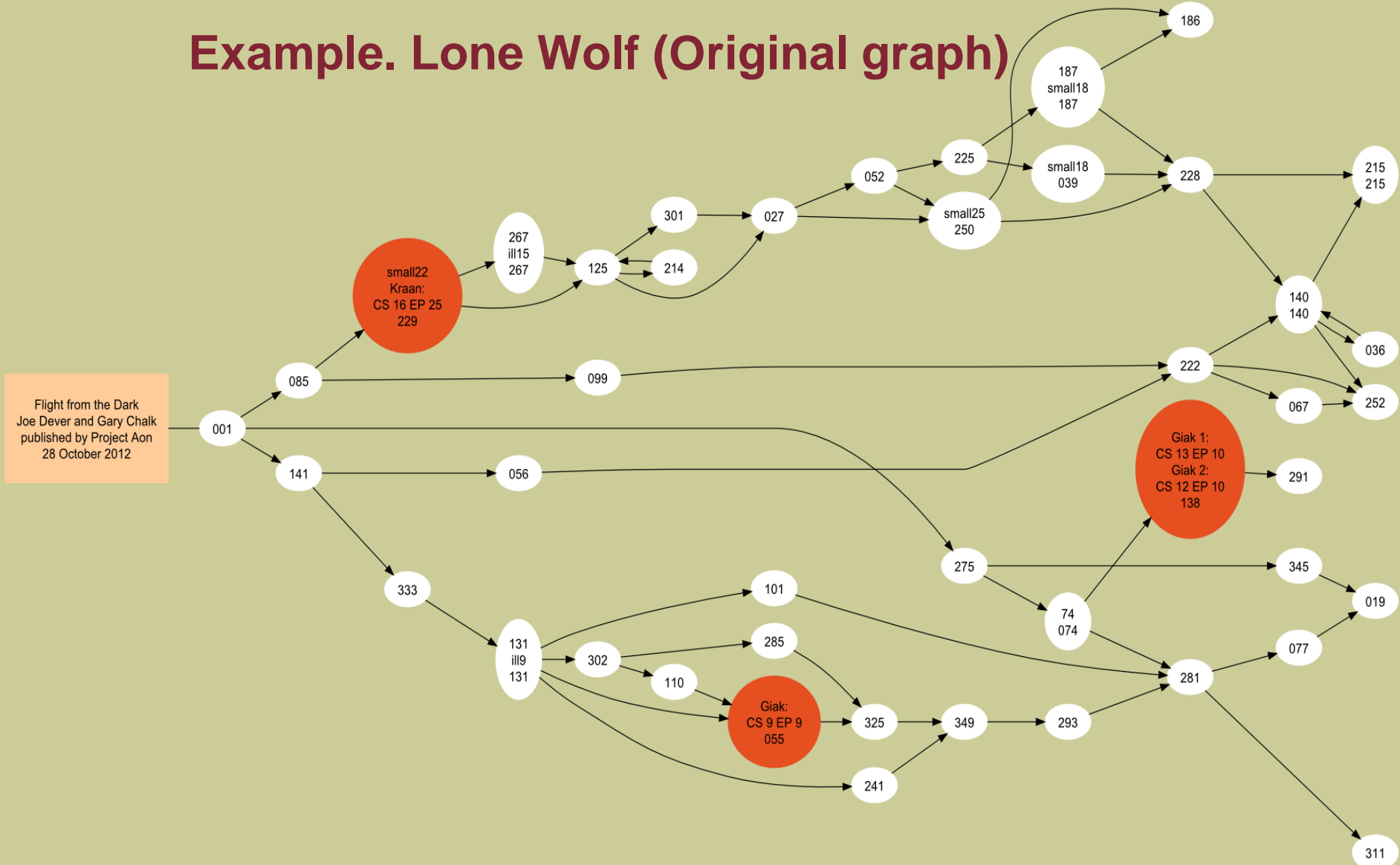
## Only small problems...

- This approach has only been tested in small examples. Now we want to investigate how it works in bigger real scenarios.

## Testbed

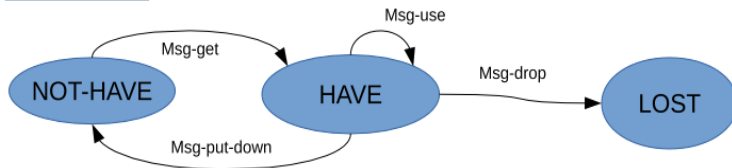
- Two JaCO representations of the GameBook *Lone Wolf* [Vol.1] “*A flight from the dark*”(J.Deever):
  - **C**hoose **Y**our **O**wn **A**dventure (CYOA) original book rules
  - Extended rules to resemble “open-world” games

# Example. Lone Wolf (Original graph)

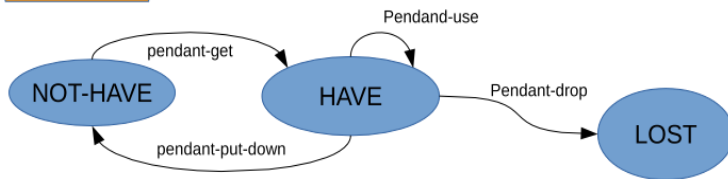


# Example. Lone Wolf (behaviors)

## MSG



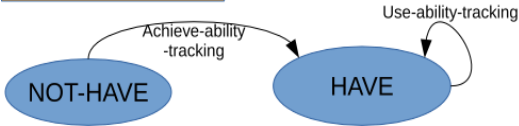
## PENDANT



## ABILITY 6°SENSE



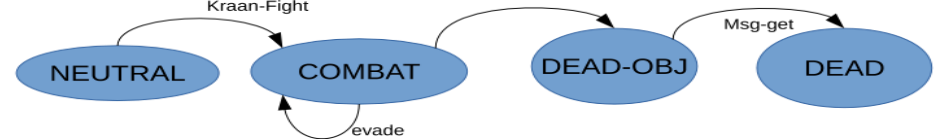
## ABILITY TRACKING



## ABILITY ANIMAL KINSHIP



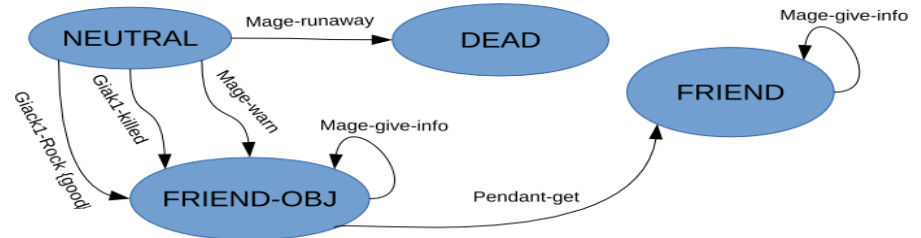
## KRAAN



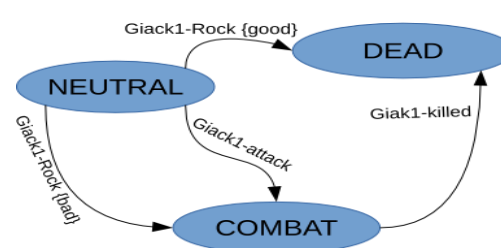
## ANIMAL



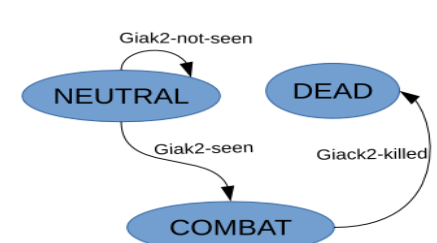
## MAGE



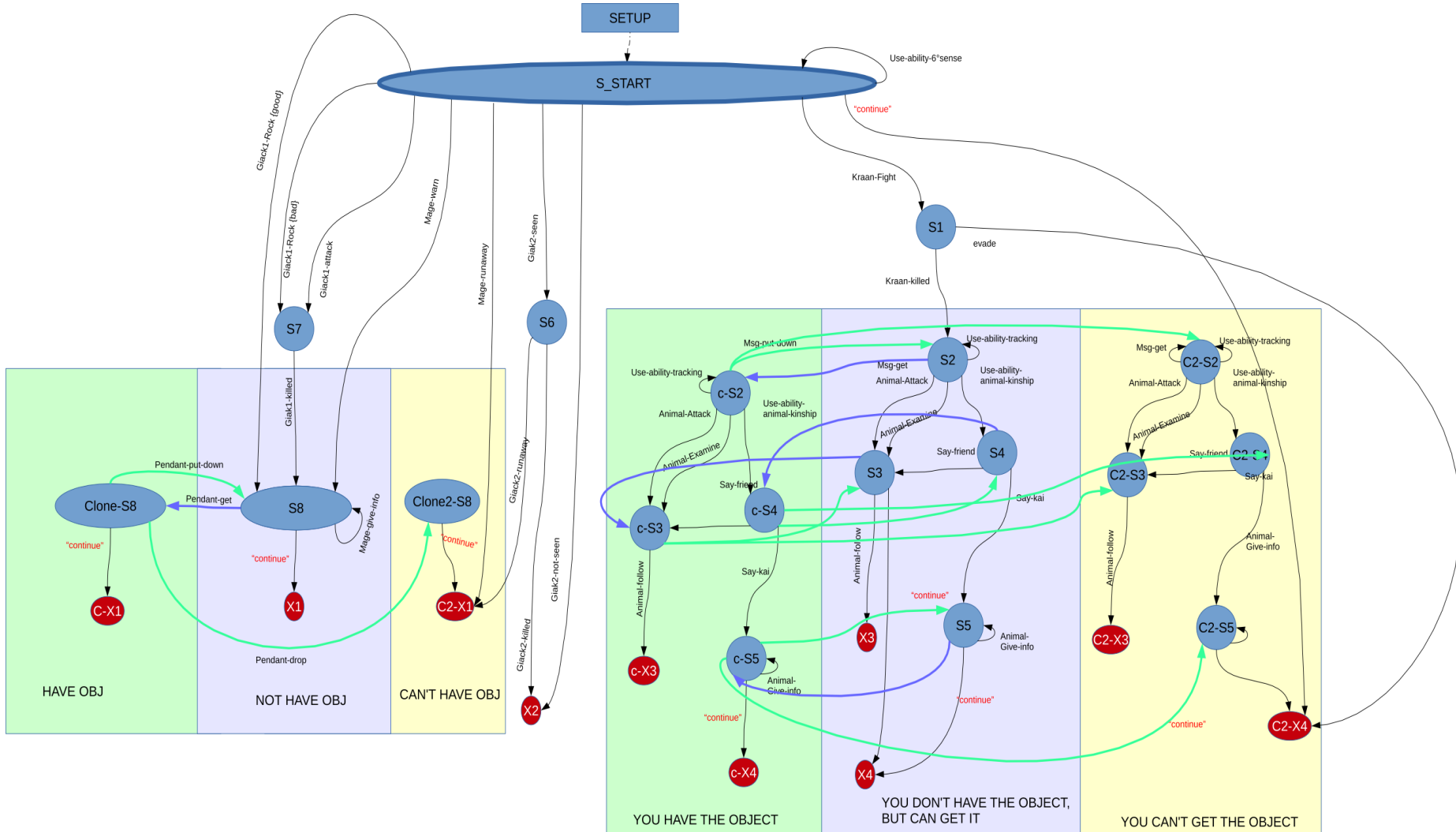
## GIACK1



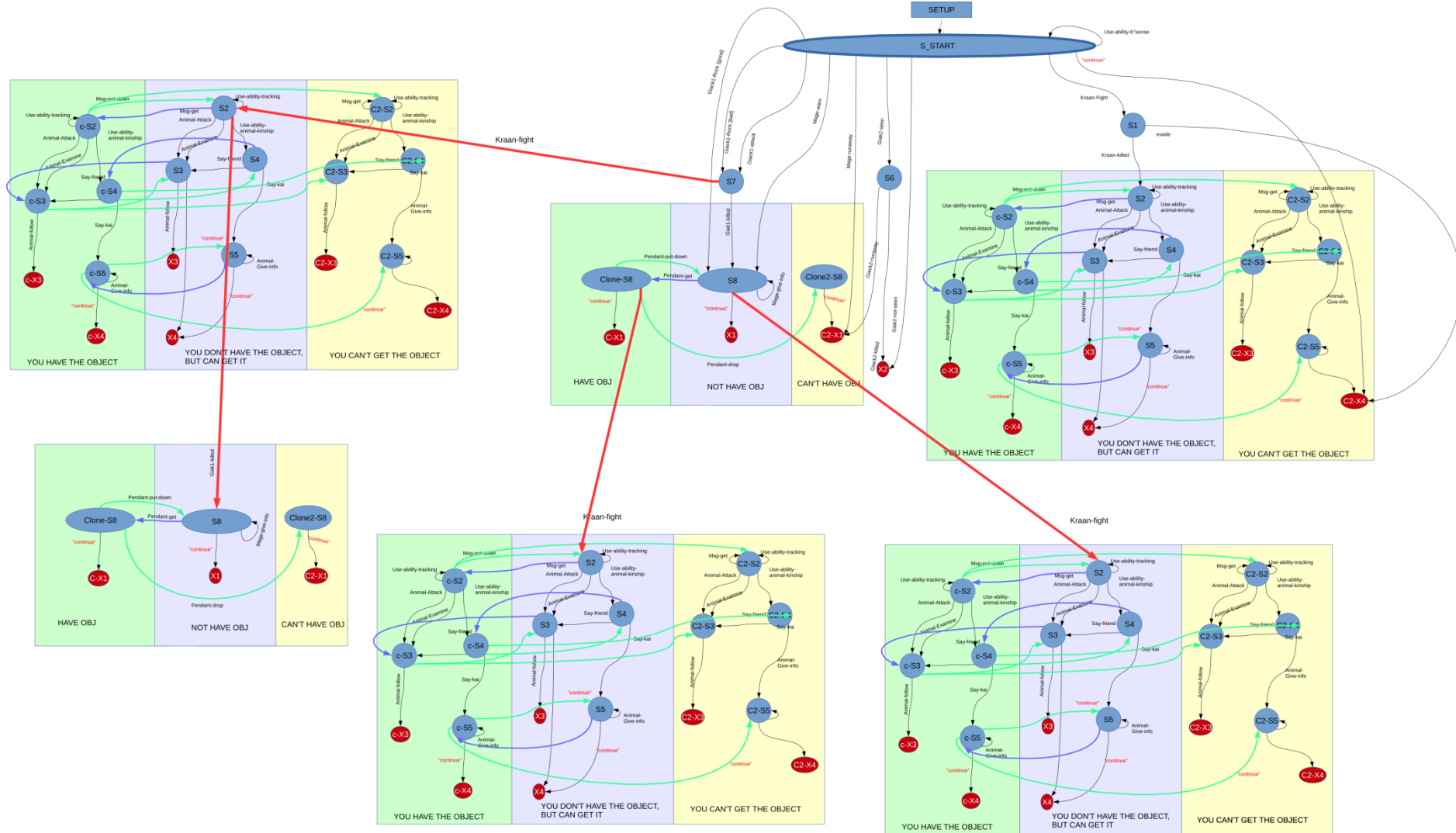
## GIACK2



### Example. Lone Wolf (CYOA target)



# Example. Lone Wolf (Unfinished Extended Target)



Videogames Story Management

## Problem of using pure transition systems

- Exponential increase of number of nodes due to uncertain or conditional transitions.
- We are using behavior composition for its automated checking, but we end up needing to do it manually

## Motivation for new approaches

- Verification of stories is a desirable capability
- A more high-level/concise modeling is needed that designers can use to formalize the intended story as a behavior composition problem



# Combining planning languages and transition systems for a new model

- **Planning Domain Definition Language (PDDL)**
  - Standard encoding language for classical *planning* tasks

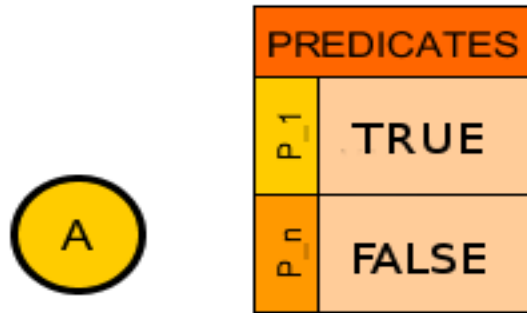
**PRO:** Transitions are concisely expressed by specifying preconditions and effects

- Solves some of the difficulty with transition systems!

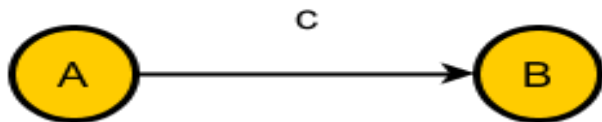
**CON:** Sequential behavior needs to be “hacked” with extra propositions

- Natural in transition systems!
- Find a representation that is close to transition systems, also incorporating some PDDL features

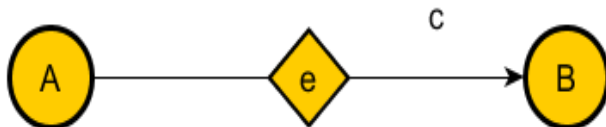
## New model components (1)



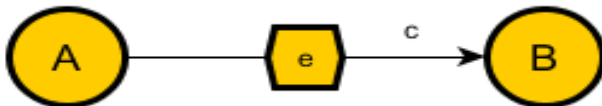
Behavior representation



Normal transition

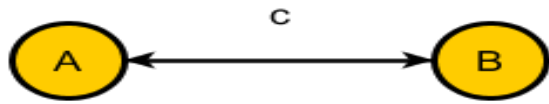


Conditioned transition

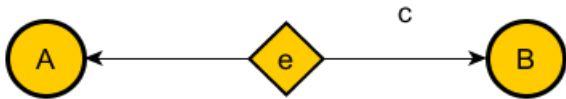


Predicate setting transition

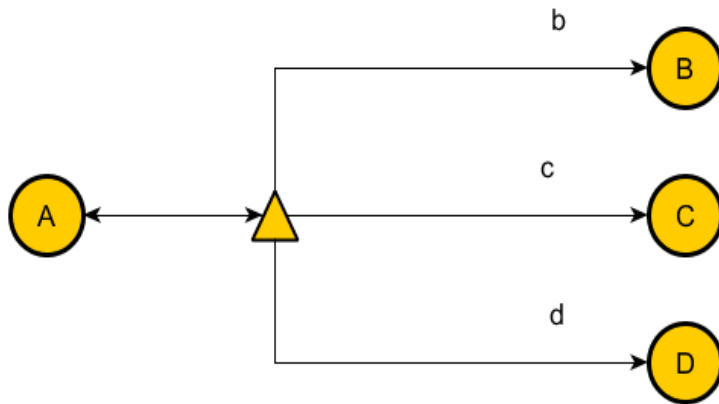
## New model components (2)



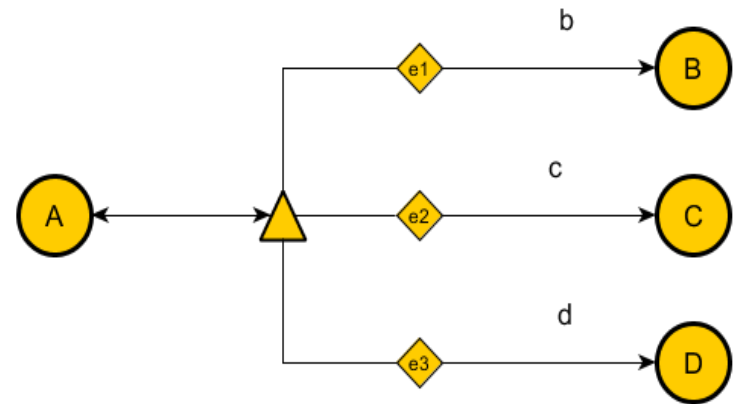
Inheriting



Conditioned inheriting

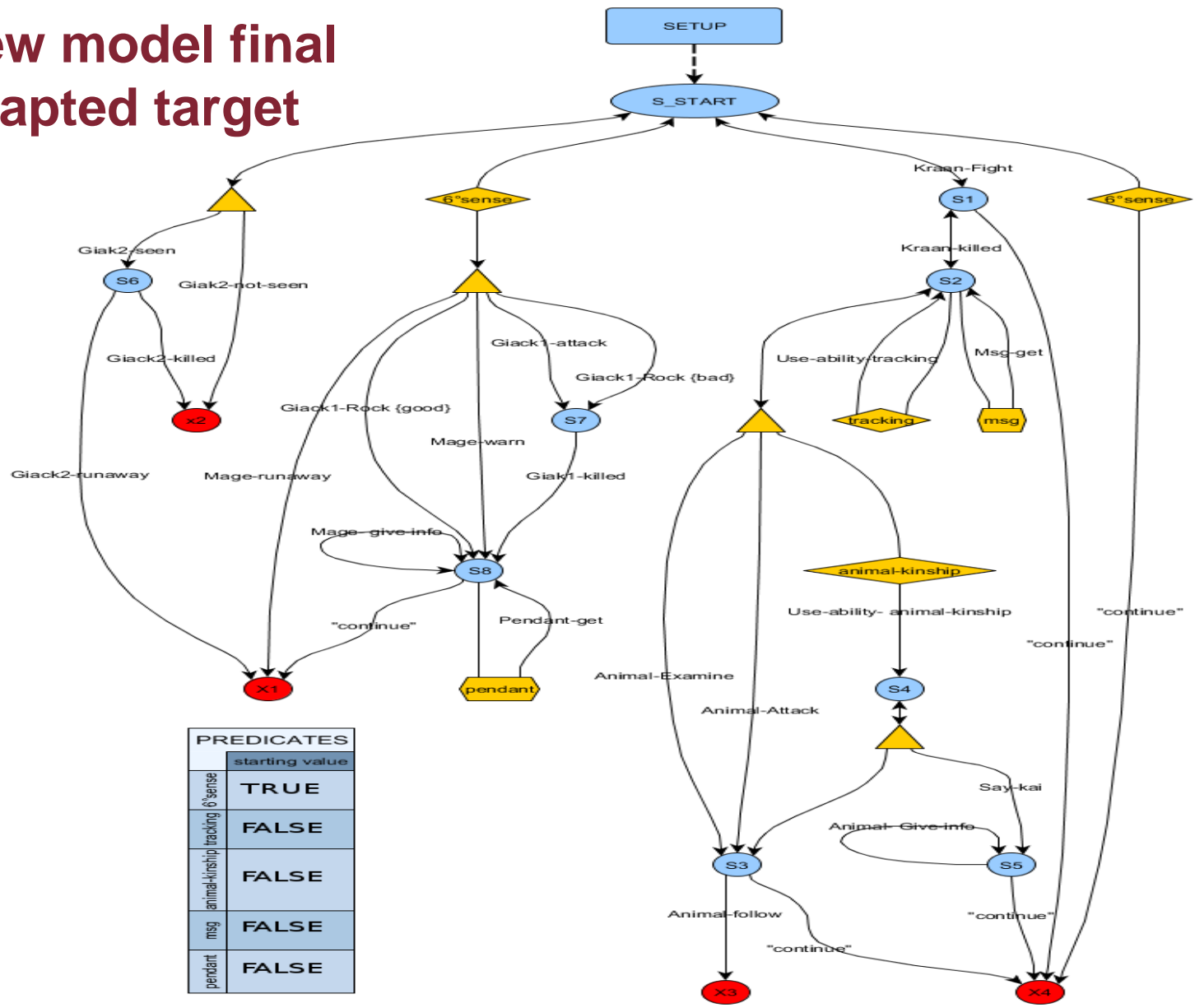


Exclusive inheriting



Conditioned exclusive inheriting

# New model final adapted target



## Conclusion and future work

- Our approach goes beyond the state-of-the-art to the direction of verified interactive storytelling
- Behavior composition relies on using generic plot points that can be facilitated by different actors
- This perspective is different than what is used in practice and needs effort to make an existing story fit in this framework
- On the other hand, behavior composition could be used as an alternative framework for building stories with different type of interactivity

## Conclusion and future work

### Future work

- Study related formalisms that feature similar “agent-program” behavior (Petri Nets, Behavior Trees, Golog) as well as default reasoning for inheritance
- Define precise semantics for the proposed graphs
- Develop a (nice!) visual tool for behavior building
- Develop plug-ins for major engines (Unity, UDK)