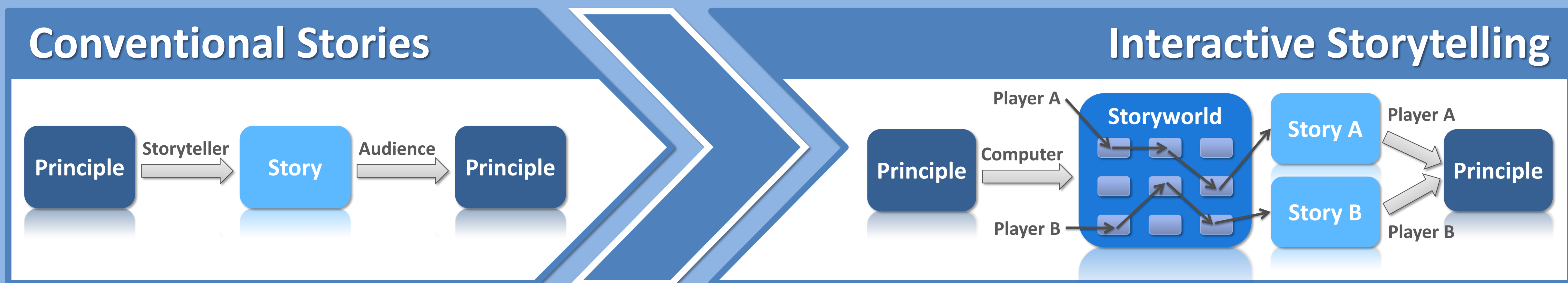# GENERATING INTERACTIVE STORIES IN COMPUTER ROLE-PLAYING GAMES

AUTHOR:
Marko Divéky, markod@nextra.sk

SUPERVISOR:
Professor **Mária Bieliková**, bielik@fiit.stuba.sk

Conventional Stories

Principle — Storyteller → Story — Audience → Principle

Interactive Storytelling

Player A
Principle — Computer → Storyworld → Story A → Player A → Principle
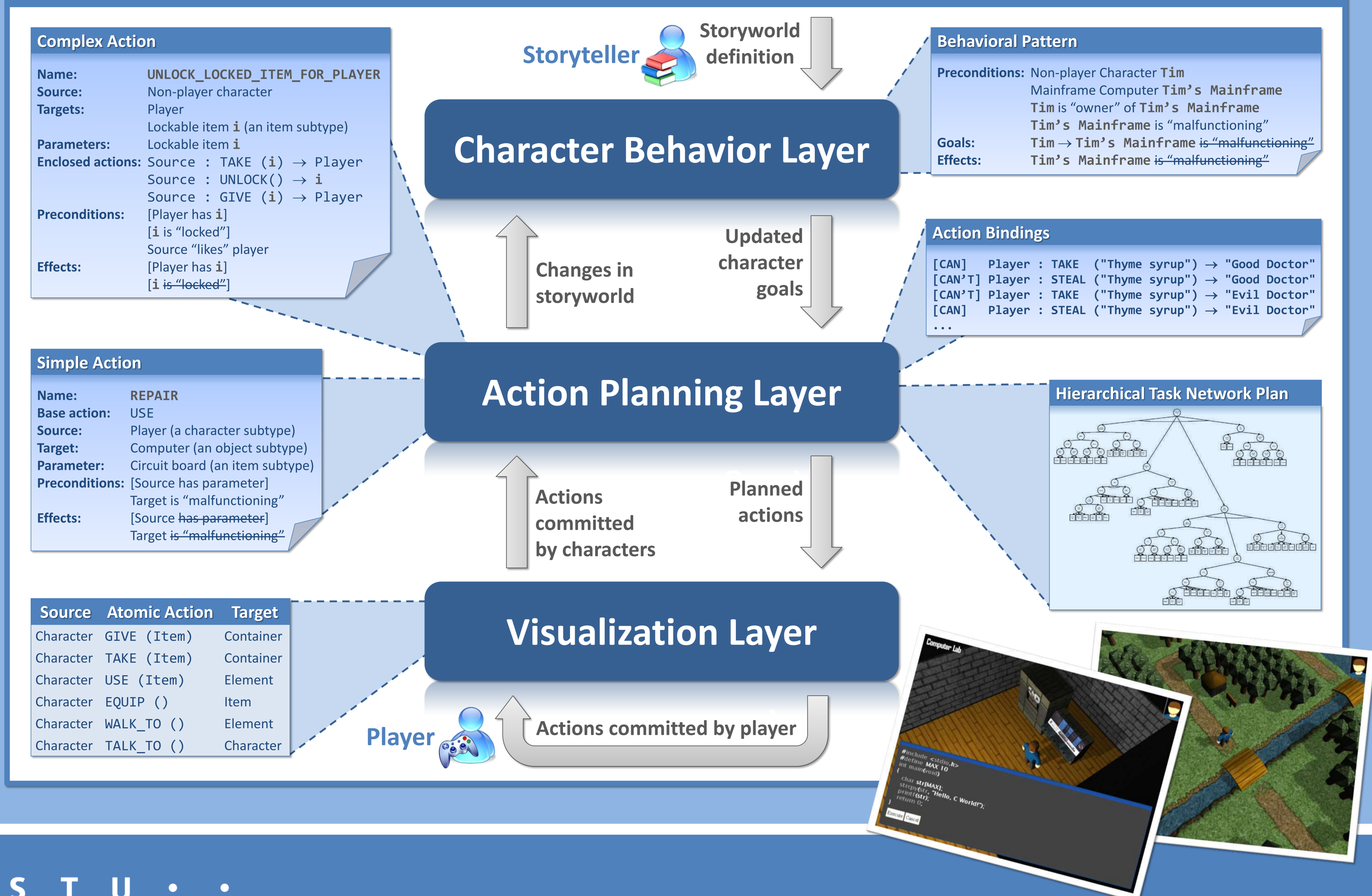Player B → Story B → Player B

Goals of This Work

✓ Visual and intuitive form of defining input data
✓ Possible to define domain-specific stories
✓ Story visualization by computer role-playing games
✓ Generated stories adapt to the player's personality

Drawbacks of Existing Solutions

✗ Complicated and unintuitive forms of defining input data
✗ Impossible to generate domain-specific interactive stories
✗ Text-based story visualization
✗ Generated stories do not adapt to player's personality

Overview of the Proposed Solution

Complex Action

Name:	UNLOCK_LOCKED_ITEM_FOR_PLAYER
Source:	Non-player character
Targets:	Player
	Lockable item i (an item subtype)
Parameters:	Lockable item i
Enclosed actions:	Source : TAKE (i) → Player
	Source : UNLOCK() → i
	Source : GIVE (i) → Player
Preconditions:	[Player has i]
	[i is "locked"]
	Source "likes" player
Effects:	[Player has i]
	[i is "locked"]

Storyteller — Storyworld definition ↓

Behavioral Pattern

Preconditions:	Non-player Character Tim
	Mainframe Computer Tim's Mainframe
	Tim is "owner" of Tim's Mainframe
	Tim's Mainframe is "malfunctioning"
Goals:	Tim → Tim's Mainframe is "malfunctioning"
Effects:	Tim's Mainframe is "malfunctioning"

Character Behavior Layer

Changes in storyworld ↑   Updated character goals ↓

Action Bindings

```
[CAN]   Player : TAKE  ("Thyme syrup") → "Good Doctor"
[CAN'T] Player : STEAL ("Thyme syrup") → "Good Doctor"
[CAN'T] Player : TAKE  ("Thyme syrup") → "Evil Doctor"
[CAN]   Player : STEAL ("Thyme syrup") → "Evil Doctor"
...
```

Simple Action

Name:	REPAIR
Base action:	USE
Source:	Player (a character subtype)
Target:	Computer (an object subtype)
Parameter:	Circuit board (an item subtype)
Preconditions:	[Source has parameter]
	Target is "malfunctioning"
Effects:	[Source has parameter]
	Target is "malfunctioning"

Action Planning Layer

Actions committed by characters ↑   Planned actions ↓

Hierarchical Task Network Plan

Source	Atomic Action	Target
Character	GIVE (Item)	Container
Character	TAKE (Item)	Container
Character	USE (Item)	Element
Character	EQUIP ()	Item
Character	WALK_TO ()	Element
Character	TALK_TO ()	Character

Visualization Layer

Player — Actions committed by player ↑