

Wide Ruled: An Author Goal-Based Interactive Story Generator

James Skorupski

expressiveintelligence**studio**

UC Santa Cruz

February 25, 2013

jskorups@soe.ucsc.edu

http://eis.ucsc.edu/Wide_Ruled

Wide Ruled



Characters

John Smith
Jerry Fontana
Cindy Rollins
Gene Franks
Rachel Delores
Dick Tracy
Sherlock Holmes

New Edit ... Delete



Environments

The Docks
The Alleyway
The City Park
The Shady Bar
The Abandoned Shack

New Edit ... Delete



Plot Point Types

Crime Info
Murderer
Murder Solved

New Edit ... Delete



Goals and Plot Fragments

- ▲ Do Murder Mystery
 - ▷ Do Story
 - No more alive victims
- ▲ Do the Crime (Text Victim)
 - ▷ Random Murderer
 - ▷ Murderer is Enemy
- ▲ Attack the Victim (Text Victim)
 - Gunshot
 - Stabbing
- ▲ Find a Detective (Text Detective)
 - Detective sleeping
 - Detective drunk
- ▲ Investigate
 - ▷ Find crime scene evidence
 - ▷ Interview friend
 - ▷ Interview coworker
 - ▷ Find clue at murderer
 - Put together clues and
- ▲ Solve the Crime
 - Solve with celebration
 - Solve with smugness
- ▲ Capture the Murderer

New ... Edit ... Delete

Wide Ruled

- Text-based interactive story generator with GUI
- Modified *UNIVERSE*-style story model (Michael Lebowitz)
 - Hierarchical, author goal based story structure
 - Originally designed for soap opera style serial storylines (never ending!)

Murderer

Name	Type
Murderer Name	Text
Murder Hideout Name	Text
Clues Found	Number

Buttons: Add New, Delete Selected, OK, Cancel

John Smith

Traits

Name	Value
Name	John Smith
Potential Victim	<input checked="" type="checkbox"/>
Age	34
Hometown	Brooklyn, New York
Gender	Male
Detective	<input type="checkbox"/>
Alive	<input checked="" type="checkbox"/>

Relationships

Name	Target	Strength
Friend	Gene Franks	10
Enemy	Jerry Fontana	3
Coworker	Cindy Rollins	10

Buttons: Edit Shared Traits ..., Edit Shared Relationships ...

UNIVERSE Overview

- Universe an **author modeling** story system
- Author plans and plot fragments that can coordinate one, many, or no characters
 - Author goals and plans may make no sense from character viewpoint
- Associated with each plot fragment
 - Author goal it can achieve
 - Characters
 - Constraints
 - Ordered list of subgoals (steps)

Example character in Universe

Name: Liz Chandler

Marriages:

Don Craig [1980]

Tony Dimera

Stereotypes: Actor, Knockout, Socialite, Party-goer

Trait modifiers: (Sex F) (Age young-adult) (Wealth 3) (Promiscuity -3)
(Intelligence 3)

Description:

Wealth 8	Promiscuity 3	Competence NIL
----------	---------------	----------------

Niceness 0	Self-Conf 6	Guile 7
------------	-------------	---------

Naiveté 7	Moodiness 6	Phys-Att 7
-----------	-------------	------------

Intelligence 7	Age young-adult	Sex F
----------------	-----------------	-------

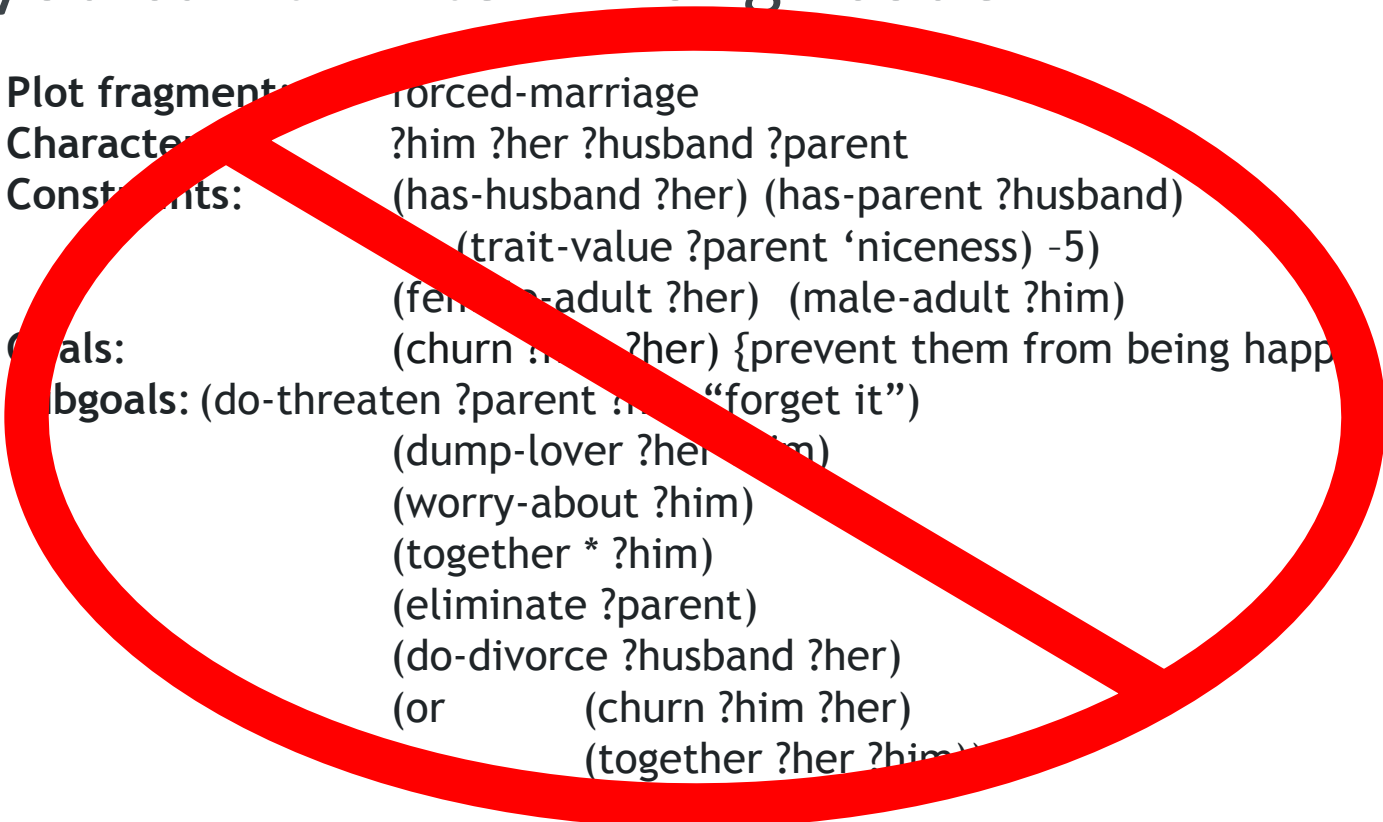
Goals: (Find-Happiness Become-Famous Meet-Famous-People)

Example plot fragment

Plot fragment: forced-marriage
Characters: ?him ?her ?husband ?parent
Constraints: (has-husband ?her) (has-parent ?husband)
(< (trait-value ?parent 'niceness) -5)
(female-adult ?her) (male-adult ?him)
Goals: (churn ?him ?her) {prevent them from being happy}
Subgoals: (do-threaten ?parent ?her "forget it")
(dump-lover ?her ?him)
(worry-about ?him)
(together * ?him)
(eliminate ?parent)
(do-divorce ?husband ?her)
(or (churn ?him ?her)
(together ?her ?him))

No more lisp

- Wide Ruled is a “structured editor” - you have access to a UNIVERSE-like story model, but you can't write “wrong” code



```
Plot fragment: forced-marriage
Character: ?him ?her ?husband ?parent
Constraints: (has-husband ?her) (has-parent ?husband)
              (trait-value ?parent 'niceness) -5)
              (female-adult ?her) (male-adult ?him)
Goals: (churn ?husband ?her) {prevent them from being happy}
Subgoals: (do-threaten ?parent ?husband "forget it")
            (dump-lover ?her ?husband)
            (worry-about ?him)
            (together * ?him)
            (eliminate ?parent)
            (do-divorce ?husband ?her)
            (or (churn ?him ?her)
                (together ?her ?him))
```

Story Components

Wide Ruled stories are made of:

- **Objects** (“What” of the story)
 - **Characters, Environments** – attributes and relationships
 - **Plot Points** – major episodic story information
- **Structure** that acts on these objects (“How” of the story)
 - **Author Goals** – What to do in the story
 - **Plot Fragments** – Ways of achieving the goals

What

The image shows a software interface with four main panels, each with a list of items and control buttons at the bottom.

- Characters** (Icon: three people):
 - John Smith
 - Jerry Fontana
 - Cindy Rollins
 - Gene Franks
 - Rachel Delores
 - Dick Tracy
 - Sherlock Holmes
- Environments** (Icon: house):
 - The Docks
 - The Alleyway
 - The City Park
 - The Shady Bar
 - The Abandoned Shack
- Plot Point Types** (Icon: lightbulb):
 - Crime Info
 - Murderer
 - Murder Solved
- Goals and Plot Fragments** (Icon: document):
 - Do Murder Mystery
 - Do Story
 - No more alive victims
 - Do the Crime (Text Victim)
 - Random Murderer
 - Murderer is Enemy
 - Attack the Victim (Text Victim)
 - Gunshot
 - Stabbing
 - Find a Detective (Text Detective)
 - Detective sleeping
 - Detective drunk
 - Investigate
 - Find crime scene evidence
 - Interview friend
 - Interview coworker
 - Find clue at murderer
 - Put together clues and
 - Solve the Crime
 - Solve with celebration
 - Solve with smugness
 - Capture the Murderer

Each panel has a 'New', 'Edit ...', and 'Delete' button at the bottom.

How

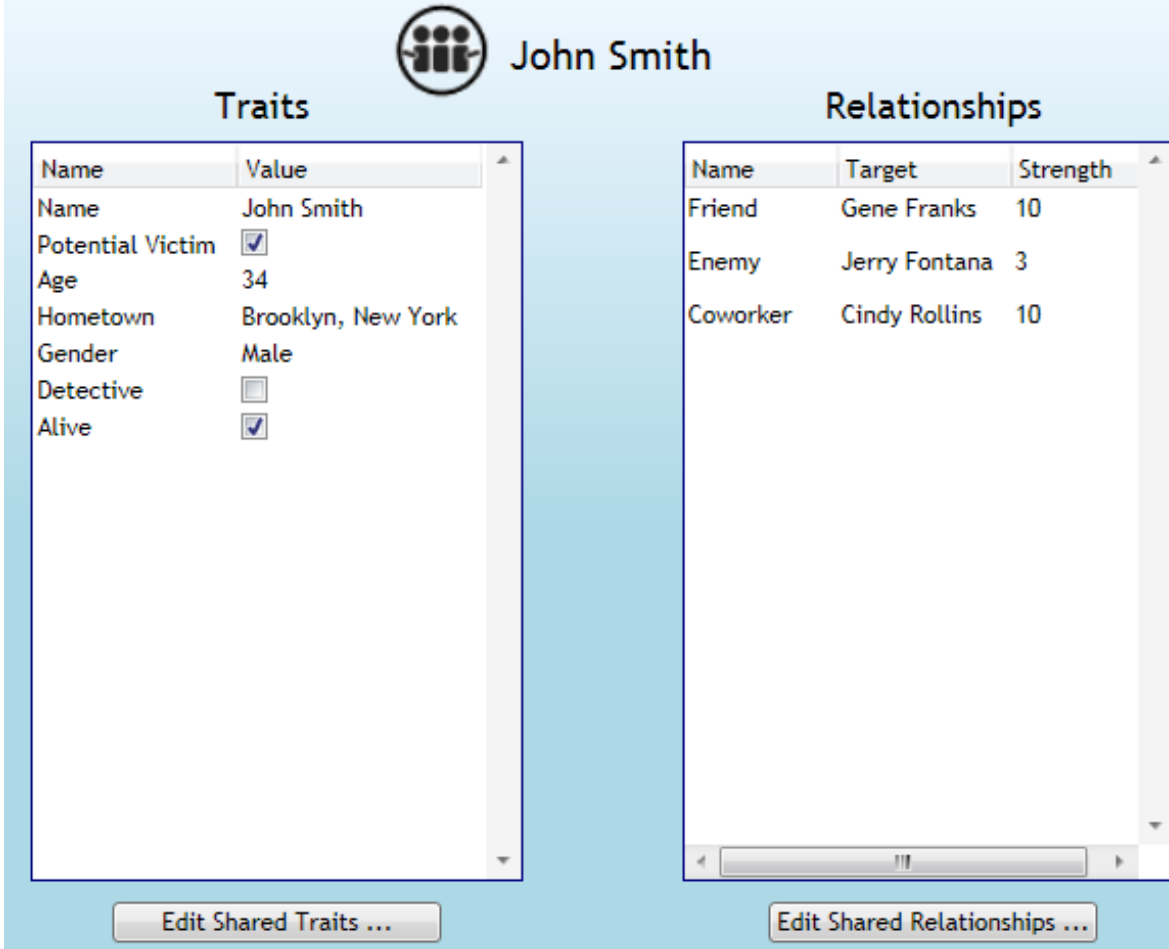
The image shows a software interface for creating a story, divided into four main sections:

- Characters:** A list of names including John Smith, Jerry Fontana, Cindy Rollins, Gene Franks, Rachel Delores, Dick Tracy, and Sherlock Holmes. Below the list are buttons for "New", "Edit ...", and "Delete".
- Environments:** A list of locations including The Docks, The Alleyway, The City Park, The Shady Bar, and The Abandoned Shack. Below the list are buttons for "New", "Edit ...", and "Delete".
- Plot Point Types:** A list of plot points including Crime Info, Murderer, and Murder Solved. Below the list are buttons for "New", "Edit ...", and "Delete".
- Goals and Plot Fragments:** A list of goals and plot fragments, including "Do Murder Mystery", "Do the Crime (Text Victim)", "Attack the Victim (Text v)", "Find a Detective (Text D)", "Investigate", "Solve the Crime", and "Capture the Murderer". Each goal has sub-items. Below the list are buttons for "New ...", "Edit ...", and "Delete".

Characters and Environments

- Object with two lists:
 - Traits (number, text, true/false)
 - Relationships (with strengths)

- Relationships connect objects of same type:
 - Character-to-character
 - Environment-to-environment



The screenshot displays a character profile for John Smith. At the top, there is a circular icon with three stylized figures and the name "John Smith". Below this, there are two main sections: "Traits" and "Relationships".

Traits

Name	Value
Name	John Smith
Potential Victim	<input checked="" type="checkbox"/>
Age	34
Hometown	Brooklyn, New York
Gender	Male
Detective	<input type="checkbox"/>
Alive	<input checked="" type="checkbox"/>

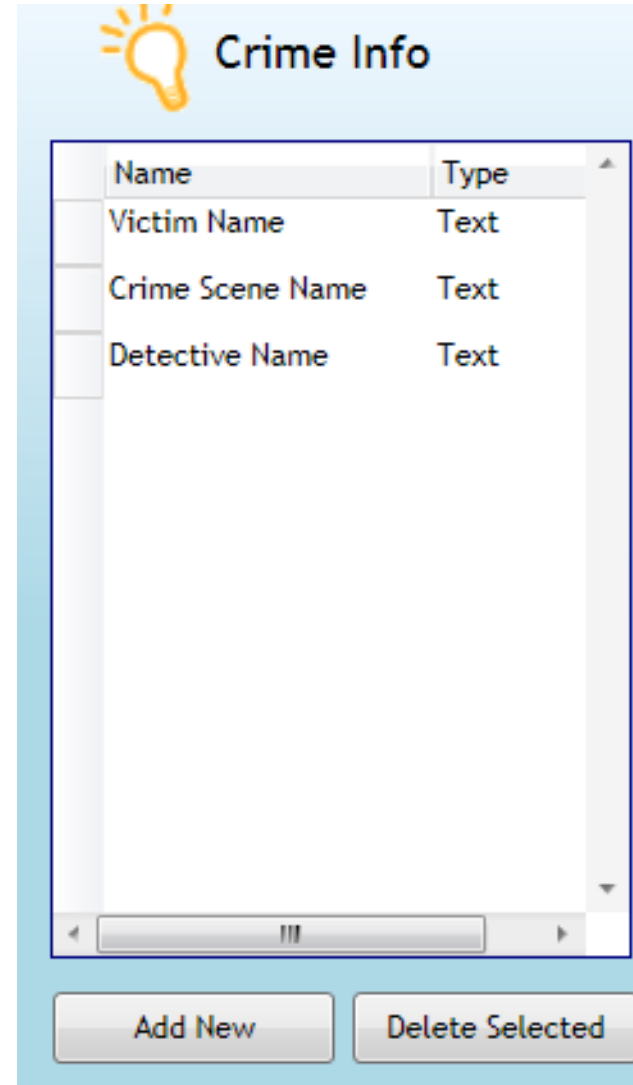
Relationships

Name	Target	Strength
Friend	Gene Franks	10
Enemy	Jerry Fontana	3
Coworker	Cindy Rollins	10

At the bottom of each section, there is a button: "Edit Shared Traits ..." and "Edit Shared Relationships ...".

Plot Points

- Plot Points are objects containing important story information:
 - Murder mystery: the murderer, the victim, the detective
- Plot Points are **temporary**, and can only be created, edited, or deleted **during story generation**
- You specify what they can contain before story generation
- ...but you fill them with information within **Plot Fragments**



The screenshot shows a user interface titled "Crime Info" with a lightbulb icon. It features a table with two columns: "Name" and "Type". The table contains three rows of data. Below the table is a horizontal scrollbar. At the bottom of the interface are two buttons: "Add New" and "Delete Selected".

Name	Type
Victim Name	Text
Crime Scene Name	Text
Detective Name	Text

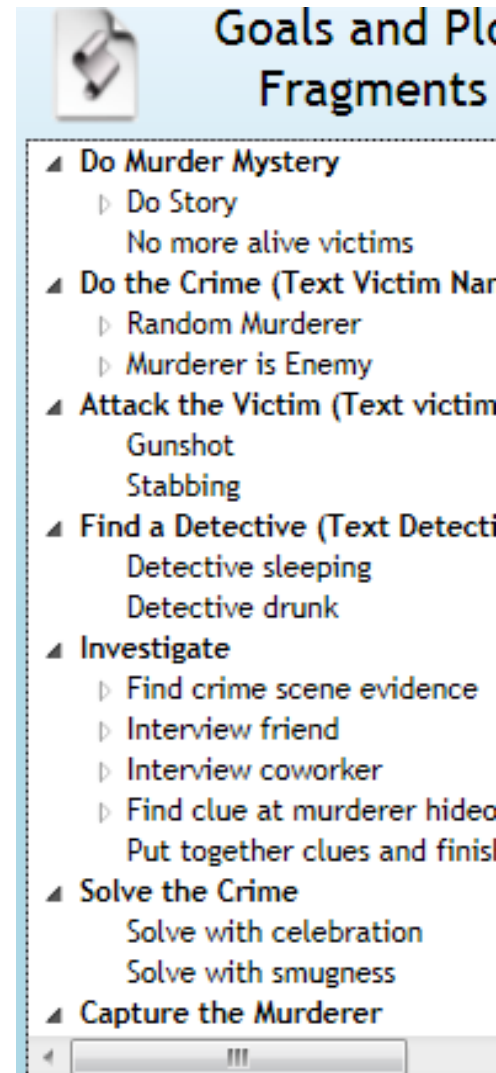
Wide Ruled

■ Author Goals

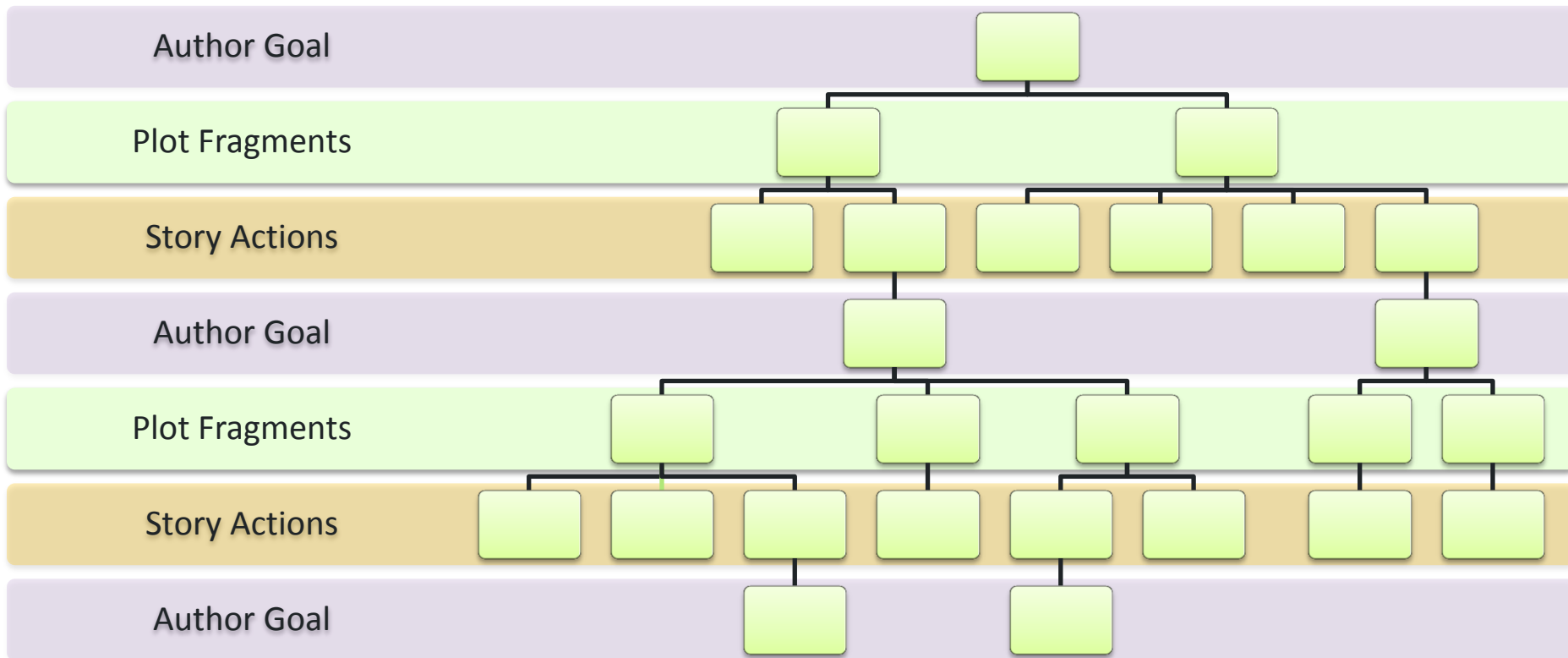
- Reflect story structure, high level intent of author
- Can involve multiple characters, environments (or none)
- **Murder mystery example:**
 “Murder Victim”, “Investigate”, “Capture criminal”

■ Plot Fragments

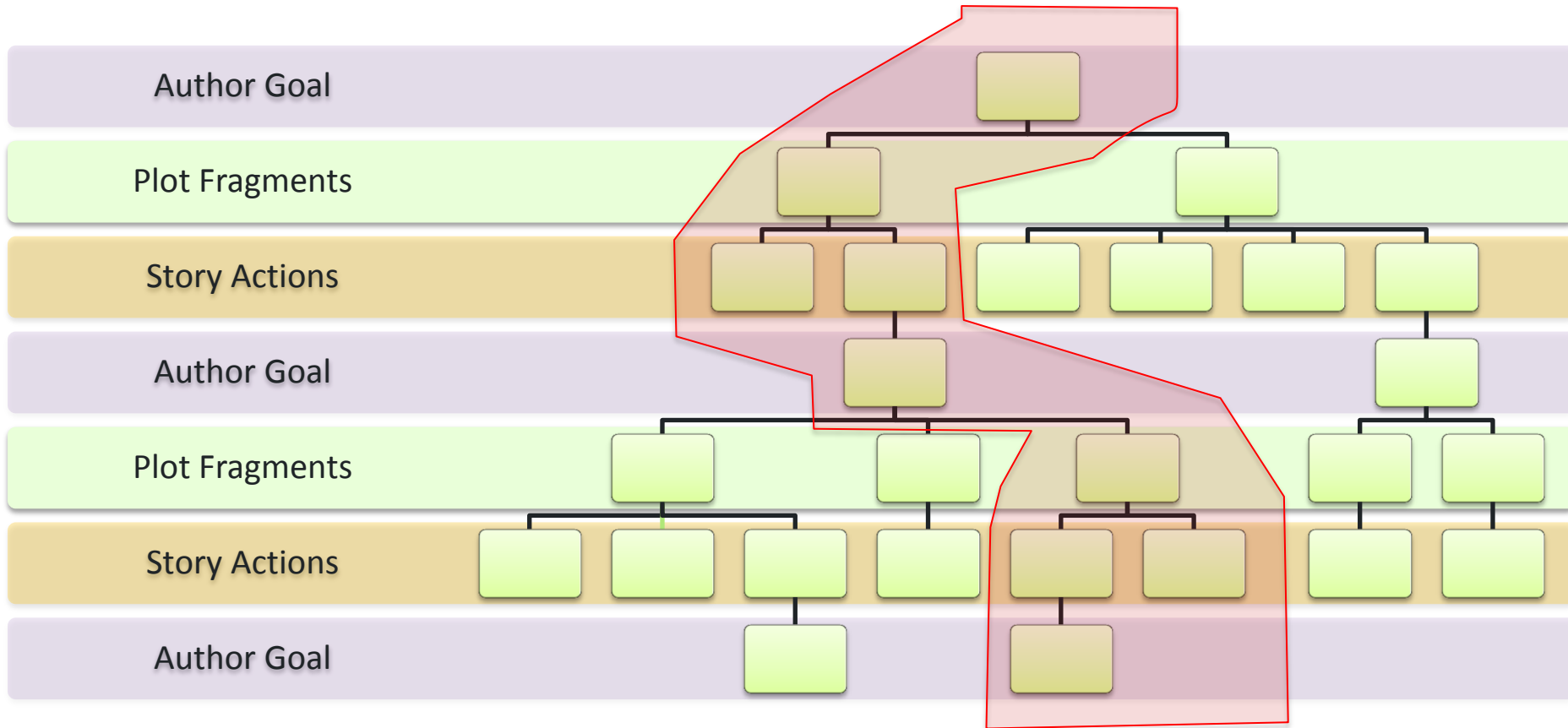
- Different ways to achieve a **single** Author Goal
 - A precondition
 - List of story actions to be completed in order
 - Can activate another author goal within it (**subgoal**)
- **Example fragments for “Investigate”:**
 “Interview family”, “Run DNA tests”, “Gather suspects”



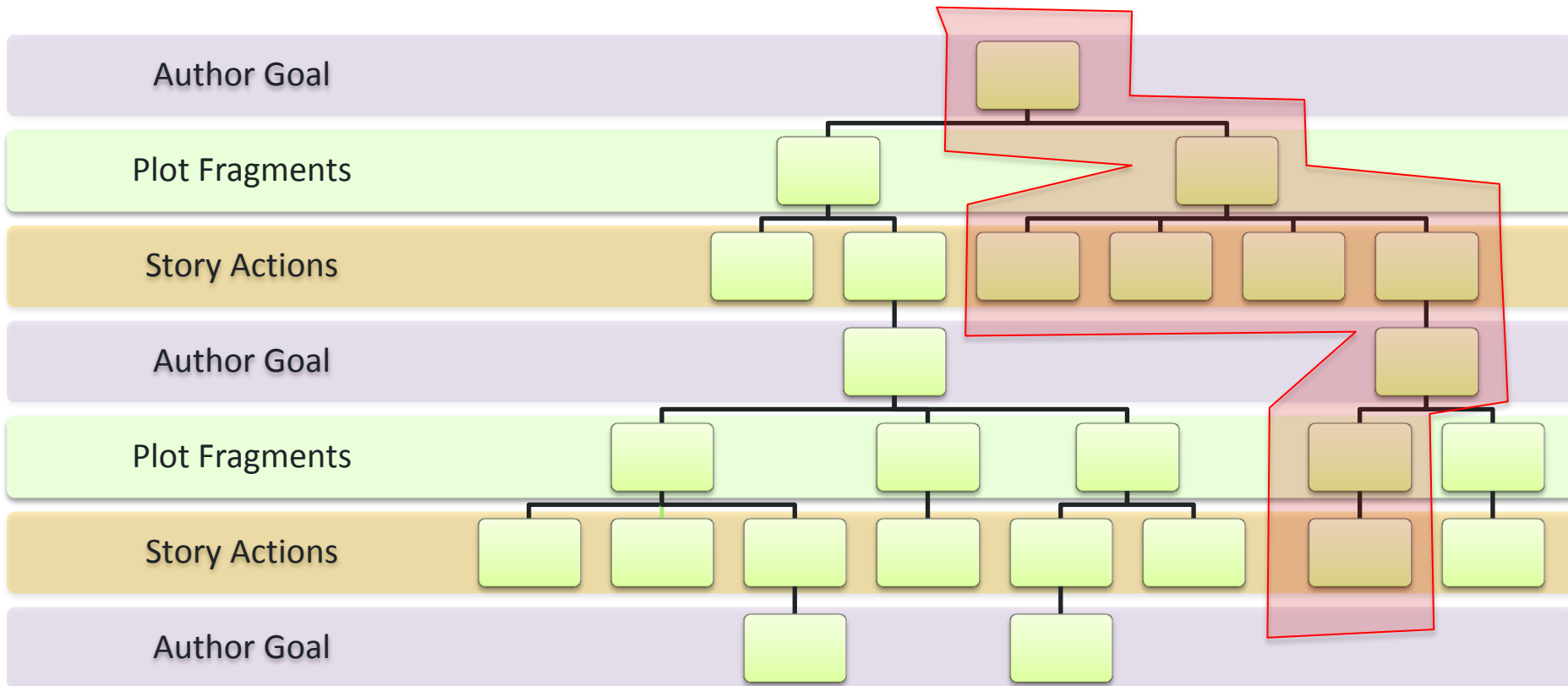
Wide Ruled



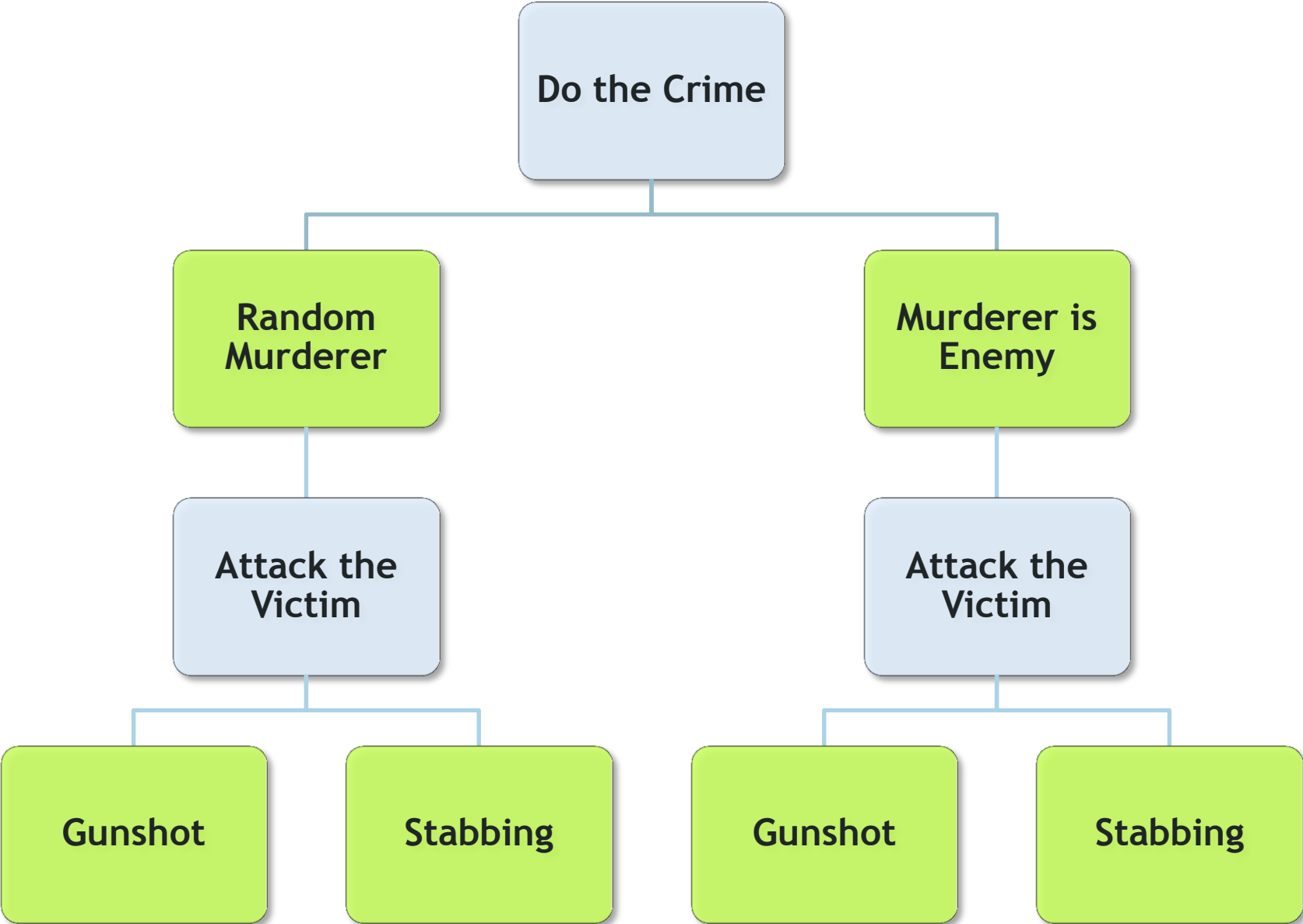
Wide Ruled



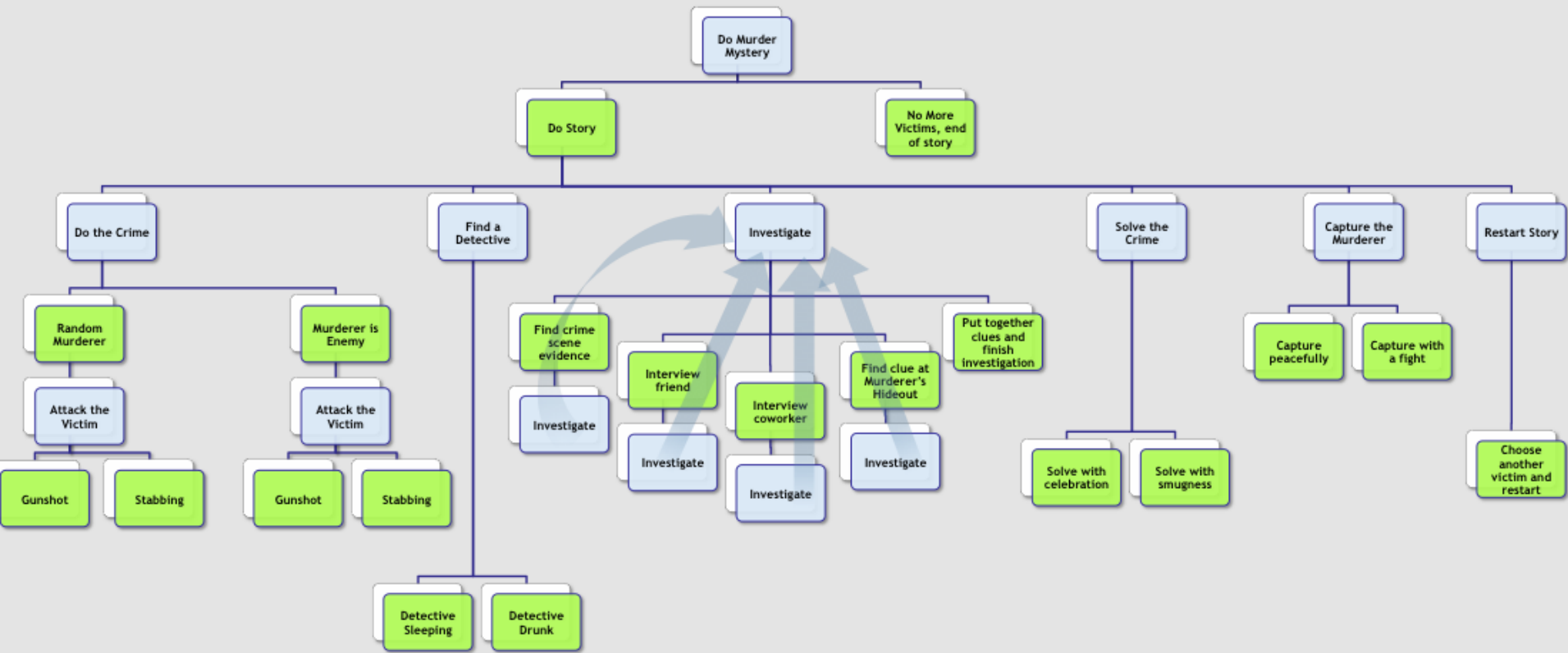
Wide Ruled



Murder Mystery Story Sample Goal



Murder Mystery Goal/Fragment Hierarchy



How do we make a single story?

- 1. Start with initial Author Goal, selected by author
- 2. Generator looks at every Plot Fragment for that Author goal
- 3. Generator checks preconditions for Plot Fragments
- 4. Generator picks **one** Plot Fragment with satisfied preconditions, and then executes every story action **in order**
 - If a story action is a **subgoal action** go to step 2

- A plot fragment can only be selected if it's entire precondition is satisfied

Author Goals and Plot Fragments - Review

■ Author Goals

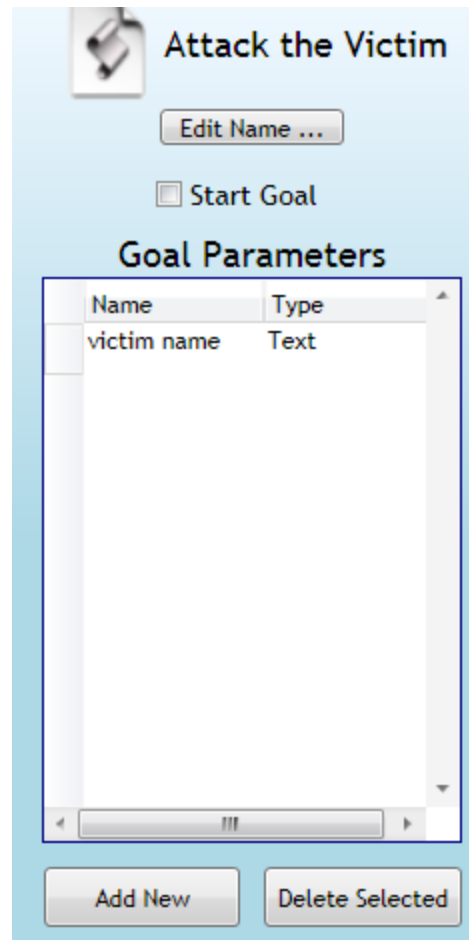
- Reflect story structure, high level intent of author

■ Plot Fragments

- Are different ways to achieve a single Author Goal
- Contain:
 - Precondition
 - Ordered list of story actions
 - print text, calculate, edit character/environment/plot point, create/delete plot point, and **subgoal**

Author Goal Editor

- Author goals can have parameters that pass information to plot fragments



Plot Fragments

- The Plot Fragment editor

The screenshot shows the Plot Fragment editor for a fragment titled "Random Murderer". The interface is divided into three main sections: Author Goal, Precondition, and Story Actions.

Author Goal: The goal is "Do the Crime". Below it, there is a "Parameters" table:

Name	Type
Victim Name	Text

Precondition: The precondition text is: "There exists a Character, where Trait 'Name' != Victim Name and Trait 'Description' is saved as variable 'crimeScene' and Trait 'Nearby Location' target name is saved as variable 'hidingPlace'".

Story Actions: The actions listed are: "Display Text: 'The time: <crimeTime>, the place: <crimeScene>'"; "Pursue Subgoal 'Attack the Victim (Victim Name)'"; "Display Text: '<murderName> appears out of the dark inevitable eyes of the police.'"; "Create new Crime Info Plot Point, saved as variable 'crimeInformation'"; "Edit saved Crime Info Plot Point 'crimeInformation': save"; "Create new Murderer Plot Point, saved as variable 'murderer'"; "Edit saved Murderer Plot Point 'murdererInformation': save"; "Edit saved Murderer Plot Point 'murdererInformation': save".

At the bottom, there are control buttons: "Change Goal ...", "New ...", "Edit ...", "Copy", "Delete", "New ...", "Edit ...", "Copy", "Delete", and "OK".

Preconditions

- A Plot Fragment is chosen if its **Precondition** is satisfied
- Each constraint statements matches against
 - Characters
 - Environments
 - Plot Points
- Precondition is a list of constraint statements that must **ALL** be true (each statement has at least one char./environ./plot point that is compatible with the constraints)

There exists a Character, where Trait "Name" == Victim Name, and Relationship "Enemy" target name is saved as variable "enemyName" and Relationship "Enemy" strength > 5

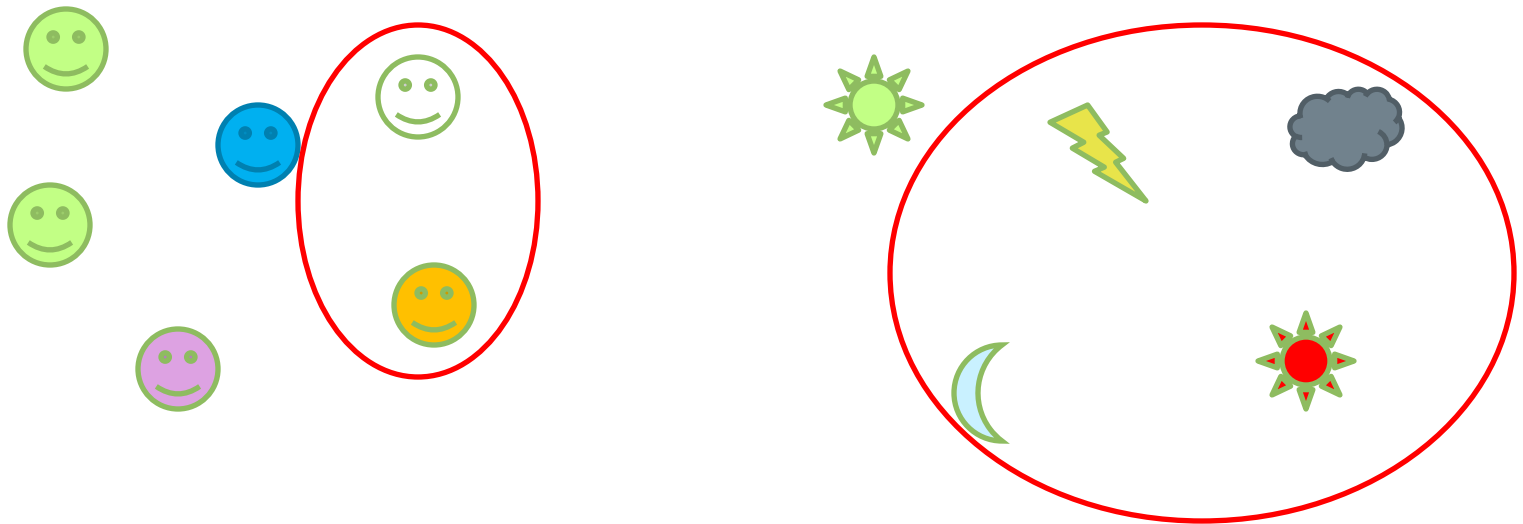
Preconditions and Story Actions

- *Preconditions capture* a subset of characters, environments, and plot points with specific attributes and relationships (character with a certain age, or environment with a certain weather, etc)



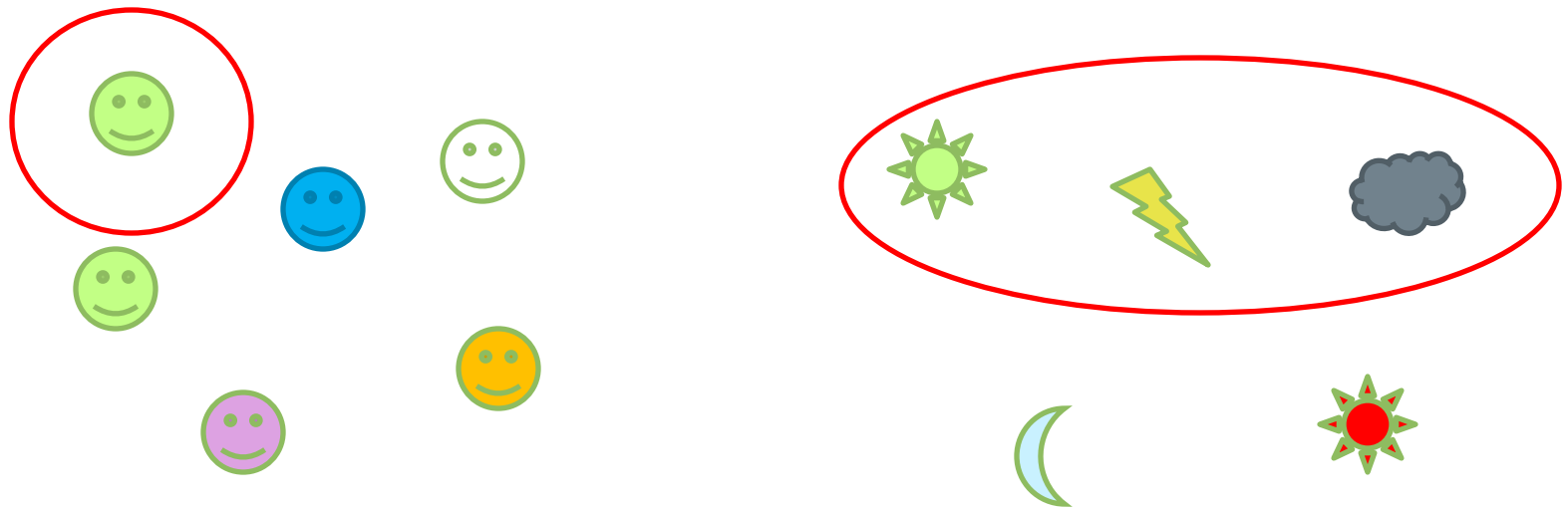
Preconditions and Story Actions

- *Preconditions capture* a subset of characters, environments, and plot points with specific attributes and relationships (character with a certain age, or environment with a certain weather, etc)



Preconditions and Story Actions

- *Preconditions capture* a subset of characters, environments, and plot points with specific attributes and relationships (character with a certain age, or environment with a certain weather, etc)



Preconditions and Story Actions

- *Preconditions capture* a subset of characters, environments, and plot points with specific attributes and relationships (character with a certain age, or environment with a certain weather, etc)



Precondition Editing

There exists a Character, where Trait "Name" == Victim Name, and Relationship "Enemy" target name is saved as variable "enemyName" and Relationship "Enemy" strength > 5

Edit Character Precondition Statement

This Character must exist

Save Character ...

Constraints

- Trait "Name" == Victim Name
- Relationship "Enemy" target name is saved as variable "enemyName"
- Relationship "Enemy" strength > 5

New Delete

Constraint Details:

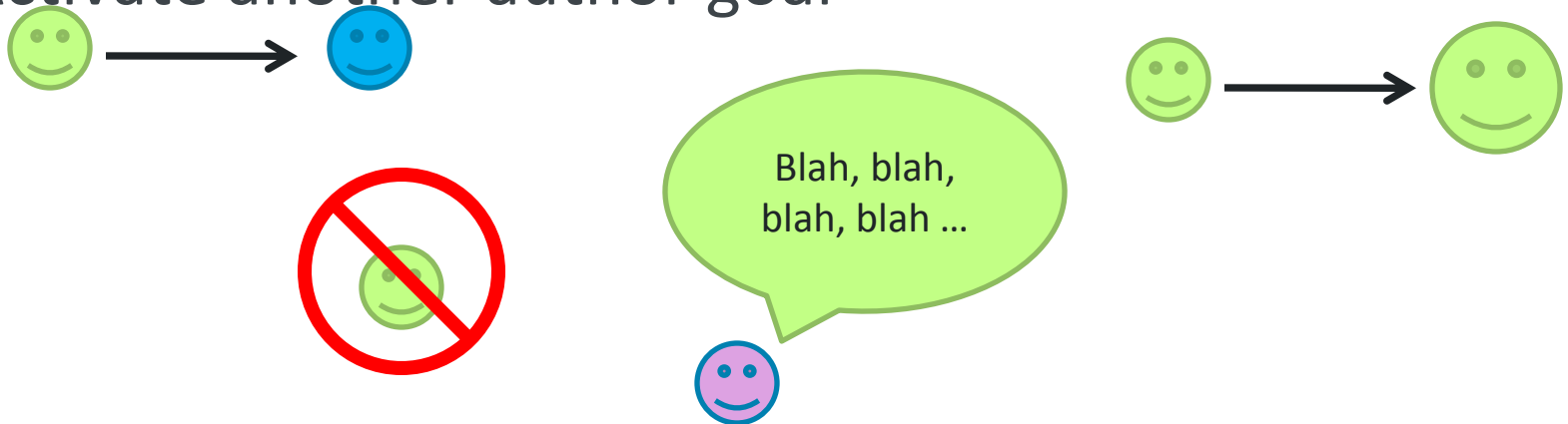
Type of attribute to match

Use Saved Variable

Must always be true

Preconditions and Story Actions

- *Story actions* **do** things with those things, for example:
 - Create a plot point and put things into it
 - Change the age/name/relationships of a character
 - Calculate a new number variable: $\text{varY} = \text{varX} + 1$
 - Print out text (how the story is made!)
 - Activate another author goal



Story Actions

■ Ordered list

Display Text: "The time: <crimeTime>, the place: <crimeSceneDescription>."

Pursue Subgoal "Attack the Victim (Victim Name)"

Display Text: "<enemyName> appears out of the darkness as the evil murderer, and escapes into the distance inevitable eyes of the police."

Create new Crime Info Plot Point, saved as variable "crimeInformation"

Edit saved Crime Info Plot Point "crimeInformation": set trait "Victim Name" to saved variable "Victim Name"

Edit saved Crime Info Plot Point "crimeInformation": set trait "Crime Scene Name" to saved variable "crimeSceneName"

Create new Murderer Plot Point, saved as variable "murderInfo"

Edit saved Murderer Plot Point "murderInfo": set trait "Murderer Name" to saved variable "enemyName"

Edit saved Murderer Plot Point "murderInfo": set trait "Murder Hideout Name" to saved variable "hideoutName"

Display Text: "<detectiveName> decides to interview <friendName>, a friend of <victimName> who was murdered. He finds that <victimName> and <murdererName> had a recent argument over money."

Calculate: $\text{newClueCount} = \text{currentClueCount} + 1$

Edit saved Murderer Plot Point "murderInfo": set trait "Clues Found" to saved variable "newClueCount"

Pursue Subgoal "Investigate"

Story Actions

- Display Text
- Pursue Subgoal (any Author Goal)
- Calculate new value
- Create Character/Environment/Plot point
- Edit Character/Environment/Plot point
- Delete Character/Environment/Plot point

Variable Binding

- Variables are pieces of saved and named data that can be used in various places in a plot fragment.
- They come from goal parameters, the precondition, and some actions (calculation, create new plot point)

The screenshot shows a plot fragment titled "Gunshot" with three main sections: Author Goal, Precondition, and Story Actions. The Author Goal section is titled "Attack the Victim" and contains a table of parameters:

Name	Type
victim name	Text

The Precondition section contains the text: "There exists a Character, saved as 'victim', where Tra". The Story Actions section contains the text: "Display Text: '<victim name> turns around suddenly a bang erupts and our victim falls flat to the ground, cc Edit saved Character 'victim': set trait 'Alive' to False". A red arrow points from the "victim name" parameter in the Author Goal section to the "<victim name>" variable in the Story Actions section, illustrating the flow of variable binding from left to right and top to bottom.

- **Variables can be used from Left to Right, top to Bottom (they must be created before using them)**

Variable Binding

- Variable referencing:

The screenshot displays the 'Gunshot' goal editor interface. It is divided into three main sections: 'Author Goal', 'Precondition', and 'Story Actions'. The 'Author Goal' section is titled 'Attack the Victim' and contains a 'Parameters' table with one entry: 'victim name' of type 'Text'. The 'Precondition' section contains the text: 'There exists a Character, saved as "victim", where Tra'. The 'Story Actions' section contains two lines of text: 'Display Text: "<victim name> turns around suddenly a bang erupts and our victim falls flat to the ground, cc' and 'Edit saved Character "victim": set trait "Alive" to False'. Red arrows indicate the flow of variable binding: one arrow points from the 'victim name' parameter in the Author Goal to the 'victim' variable in the Precondition, and another arrow points from the '<victim name>' placeholder in the Story Actions back to the 'victim' variable in the Precondition. Below each section are control buttons: 'Change Goal ...' for the Author Goal, and 'New ...', 'Edit ...', 'Copy', and 'Delete' for both the Precondition and Story Actions sections.

Gunshot
Edit Name ...

Author Goal
Attack the Victim

Parameters

Name	Type
victim name	Text

Precondition
There exists a Character, saved as "victim", where Tra

Story Actions
Display Text: "<victim name> turns around suddenly a bang erupts and our victim falls flat to the ground, cc
Edit saved Character "victim": set trait "Alive" to False

Change Goal ... New ... Edit ... Copy Delete New ... Edit ... Copy Delete

Variable Binding – Precondition

- Precondition Statements can save traits, relationship target names, and relationship strength values to variables for later use
 - Save trait: There exists an Environment, where Trait “Description” is saved as variable “crimeSceneDescription”
 - Save entire object: There exists a Character, saved as “victim”.
- Create a variable and then use it again for comparison
 - Example: Match two characters with different names, and one is younger than the other. Two precondition statements, in order:
 - 1. There exists a Character, where Trait name is saved as variable “**nameVar**”, and Trait Age is saved as variable ‘**friendAge**’
 - 2. There exists a Character, where Trait name **!= nameVar**, and Trait Age **< friendAge**

Variable Binding – Precondition

Edit Character Precondition Statement
This Character must exist

Save Character ... (circled in red)

Constraints

Trait "Name" == victim name

New Delete

Constraint Details:

Type of attribute to match

Use Saved Variable

Must always be true

Edit Environment Precondition Statement
This Environment must exist

Save Environment ...

Constraints

Trait "Potential Murder Location" == True
Trait "Name" ==
Trait "Description" is saved as variable "crimeSceneDescription"
Trait "Time" is saved as variable "crimeTime"
Relationship "Nearby Location" target name is saved as variable

New Delete

Constraint Details:

Type of attribute to match Trait

Description Save Only (circled in red)

Value saved as: crimeSceneDescription Use Saved Variable

Edit this Variable Must always be true

Edit Environment Precondition Statement
This Environment must exist

Save Environment ...

Constraints

Trait "Potential Murder Location" == True
Trait "Name" !=
Trait "Description" is saved as variable "crimeSceneDescription"
Trait "Time" is saved as variable "crimeTime"
Relationship "Nearby Location" target name is saved as variable "hic

New Delete

Constraint Details:

Type of attribute to match Trait

Name != Bob

Save this Value .. (circled in red) Must always be true

Variable Binding

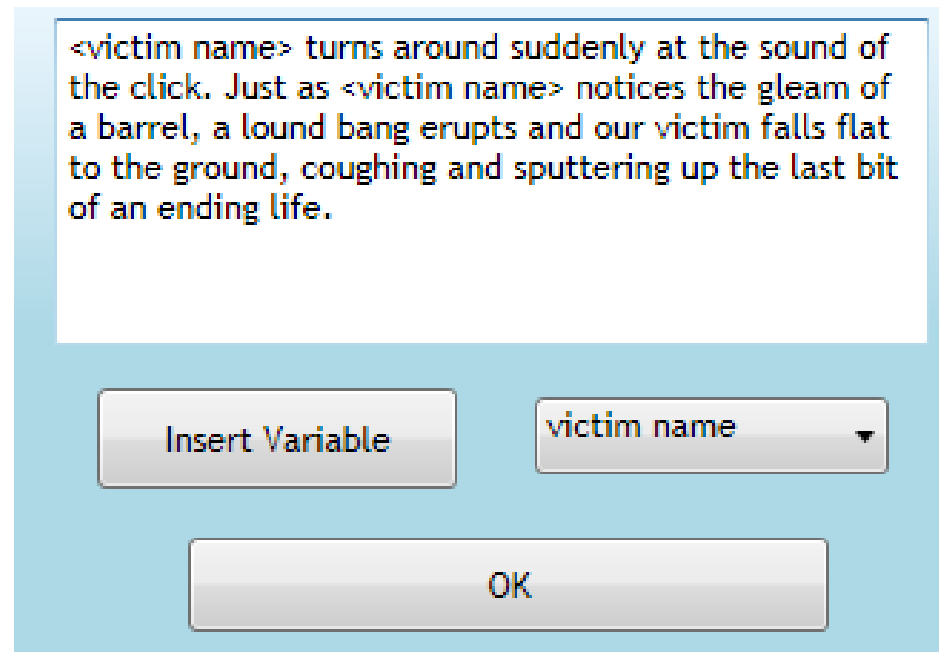
The screenshot displays the 'Gunshot' scenario editor interface. It is divided into three main sections: Author Goal, Precondition, and Story Actions.

- Author Goal:** Titled 'Attack the Victim', it includes a 'Parameters' table with one entry: 'victim name' of type 'Text'. A 'Change Goal ...' button is located below the table.
- Precondition:** Contains the text 'There exists a Character, saved as "victim", where Tra'. The variable 'victim' is circled in red. Below the text area are 'New ...', 'Edit ...', 'Copy', and 'Delete' buttons.
- Story Actions:** Contains two actions: 'Display Text: "<victim name> turns around suddenly a bang erupts and our victim falls flat to the ground, cc' and 'Edit saved Character "victim": set trait "Alive" to False'. The second action is circled in red. Below the text area are 'New ...', 'Edit ...', 'Copy', and 'Delete' buttons.

A red arrow points from the circled 'victim' in the Precondition section to the circled 'victim' in the Story Actions section, illustrating the variable binding.

Making the story appear

- **Displaying Text** - printed text that can contain captured information (variables) from the rest of the plot fragment



Making the story appear

- The time: 4 am, the place: the large, open city park in the middle of the city. Rachel Delores turns around suddenly at the sound of the click. Just as Rachel notices the gleam of a barrel, a loud bang erupts and our victim falls flat to the ground, coughing and sputtering up the last bit of an ending life. Jerry Fontana appears out of the darkness as the evil murderer, and escapes into the distance to seek shelter from the inevitable eyes of the police. Cut to a quiet single bedroom apartment. The shrill ring of a phone awakens Sherlock Holmes from a deep and relentless slumber ...

Making the story appear

- The time: 9 pm, the place: a dark rain-drenched alley off 15th street. Kim Bates turns around suddenly at the sound of the click. Just as Kim notices the gleam of a barrel, a loud bang erupts and our victim falls flat to the ground, coughing and sputtering up the last bit of an ending life. Fred Gordon appears out of the darkness as the evil murderer, and escapes into the distance to seek shelter from the inevitable eyes of the police. Cut to a quiet single bedroom apartment. The shrill ring of a phone awakens Dick Tracy from a deep and relentless slumber ...

Plot Fragments



Random Murderer

Edit Name ...

Author Goal

Do the Crime

Parameters

Name	Type
Victim Name	Text

Change Goal ...

Precondition

There exists a Character, where Trait "Name" != Victim
There exists an Environment, where Trait "Potential M
and Trait "Description" is saved as variable "crimeScen
"Nearby Location" target name is saved as variable "hi

New ...

Edit ...

Copy

Delete

Story Actions

Display Text: "The time: <crimeTime>, the place: <cr
Pursue Subgoal "Attack the Victim (Victim Name)"
Display Text: "<murderName> appears out of the dark
inevitable eyes of the police."
Create new Crime Info Plot Point, saved as variable "c
Edit saved Crime Info Plot Point "crimeInformation": s
Edit saved Crime Info Plot Point "crimeInformation": s
Create new Murderer Plot Point, saved as variable "m
Edit saved Murderer Plot Point "murderInformation": s
Edit saved Murderer Plot Point "murderInformation": s

New ...

Edit ...

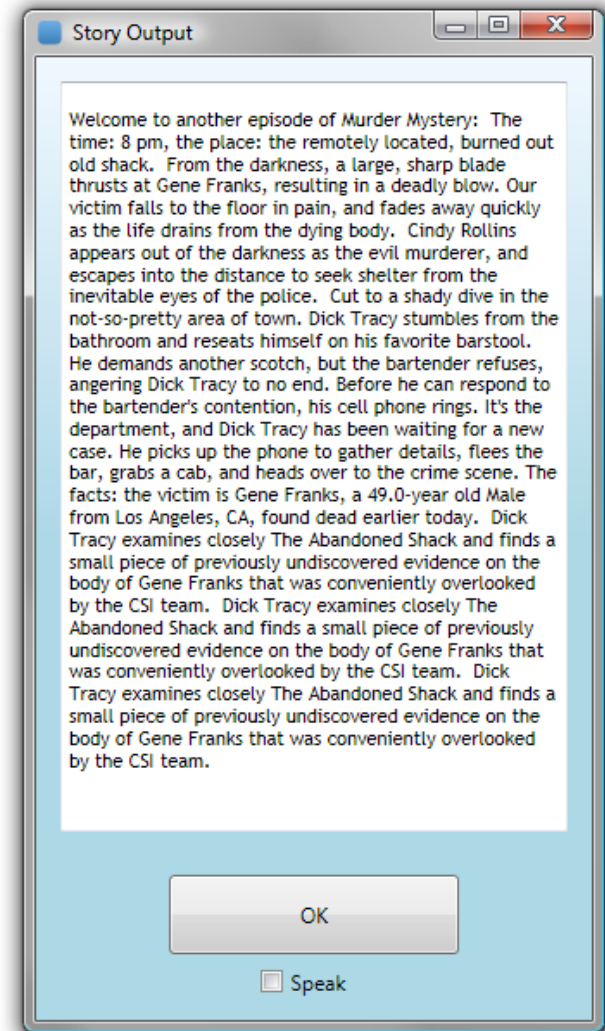
Copy

Delete

OK

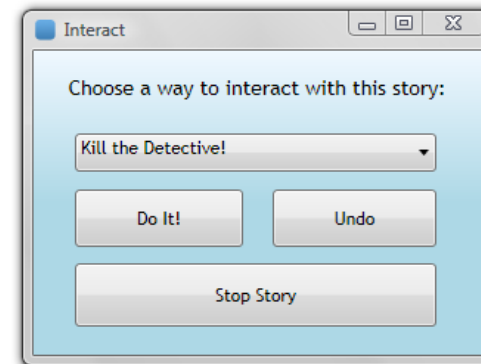
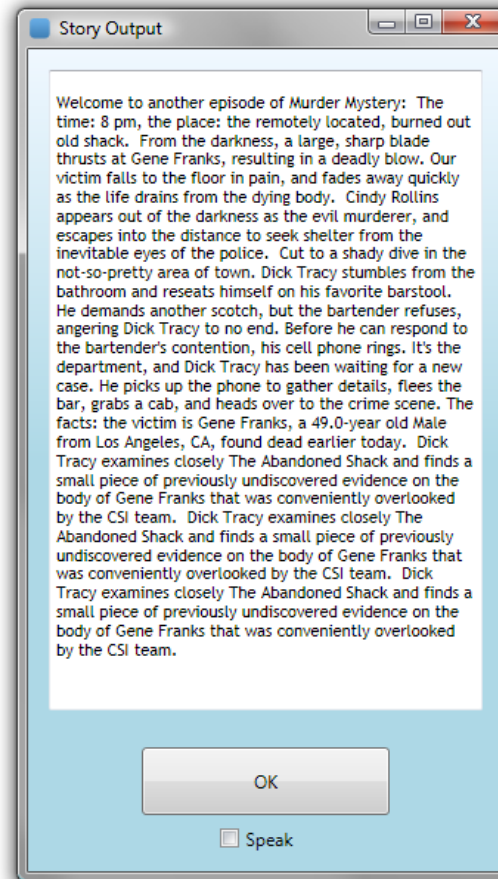
Reading

- A single story is generated on the fly, to a text box, every time you click “Generate Story”
- It is slowed down, so readers can interact with it (next slide)



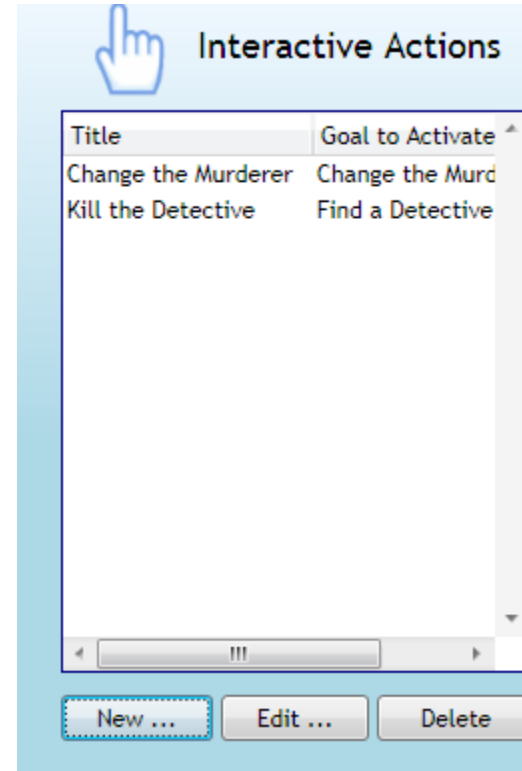
Interactivity

- Interactive Actions
 - Third person, Asynchronous
 - Author-specified **goals** which reader can activate **at any time**
 - Reader can **interrupt** the story at any time and perform meaningful, author-designed actions
 - Murder mystery examples: “Kill the detective”, “Change the murderer”, etc



Interactivity

- What are “Interactive Actions”?
- Just regular author goals with plot fragments, that the author allows to be activated during the reading of the story
- These goals are executed completely before continuing with the previous story
 - Any changes to the story world are kept
 - All subgoals are activated (this could create very long “interruptions” of the story)



Interactivity

.....He accepts his fate quickly, picks up the phone, and gathers the details about his new case. Once the call is finished, he jumps in the car, and heads to the crime scene. The facts: the victim is Rachel Delores, a 51-year old woman from Lyndhurst, New Jersey, found dead earlier today.

Cut to the dark hideout of our ruthless killer. Before sleeping for the night, **Jerry Fontana** stares coldly into a bathroom mirror and removes a thin latex mask. Underneath the disguise appears the face of **John Smith**, the true villain in this story.

Sherlock Holmes decides to interview **John Smith**, a friend of Rachel, for any clues as to why our victim was murdered. He finds that Rachel and John had a recent argument over money ...

Misc Plot Fragment Notes

- Must select **start goal**: use “Start Goal” check box, can’t have any parameters (by default, start goal is first one you create)
- **Creating** char’s/env’s/plot points within a Plot Fragment:
 - **1. New Action** “Create New ...”, save as a variable name, fill it with static numbers or text
 - **2. Edit Action**: To insert variables, perform edit action on char/env/pp saved in creation action,
- **Deleting** char’s/env/plot points within Plot Fragment
 - 1. Match in precondition, save to variable
 - 2. Use **Delete Action** on variable
- **Editing** char’s/env’s/plot points
 - 1. Match in precondition, save to variable
 - 2. For **every** trait/relationship to edit, you create **one Edit Action**, and select that saved variable in step 1.

The screenshot displays the 'Do Murder Mystery' plot fragment configuration in the Expressive Intelligence Studio. At the top, there is a title bar with a document icon and the text 'Do Murder Mystery'. Below the title bar is an 'Edit Name ...' button and a checked 'Start Goal' checkbox. Underneath is a 'Goal Parameters' section with a table that has two columns: 'Name' and 'Type'. The table is currently empty. Below the table is a 'Story Act' section with a list of actions and their display text. The actions include 'Display Text', 'Pursue Subgoal', 'Create new Crime Info Plot Point', 'Edit saved Crime Info Plot Point', 'Create new Murderer Plot Point', and 'Edit saved Murderer Plot Point'. At the bottom of the interface are three buttons: 'New ...', 'Edit ...', and 'Cancel'.

Notes

- Website: [http://eis.ucsc.edu/Wide Ruled](http://eis.ucsc.edu/Wide_Ruled) (Click on Project Page link for download/installation info)
- Tutorial posted online – Step by step instructions
- Documentation, sample story online, with these slides
- Wide Ruled is Windows XP, Vista, 7 only
 - Installation instructions posted online
 - Uses .NET 3.5 – need version 3.5 or later
 - Works with Parallels, VMWare Fusion on OS X, VirtualBox (probably)

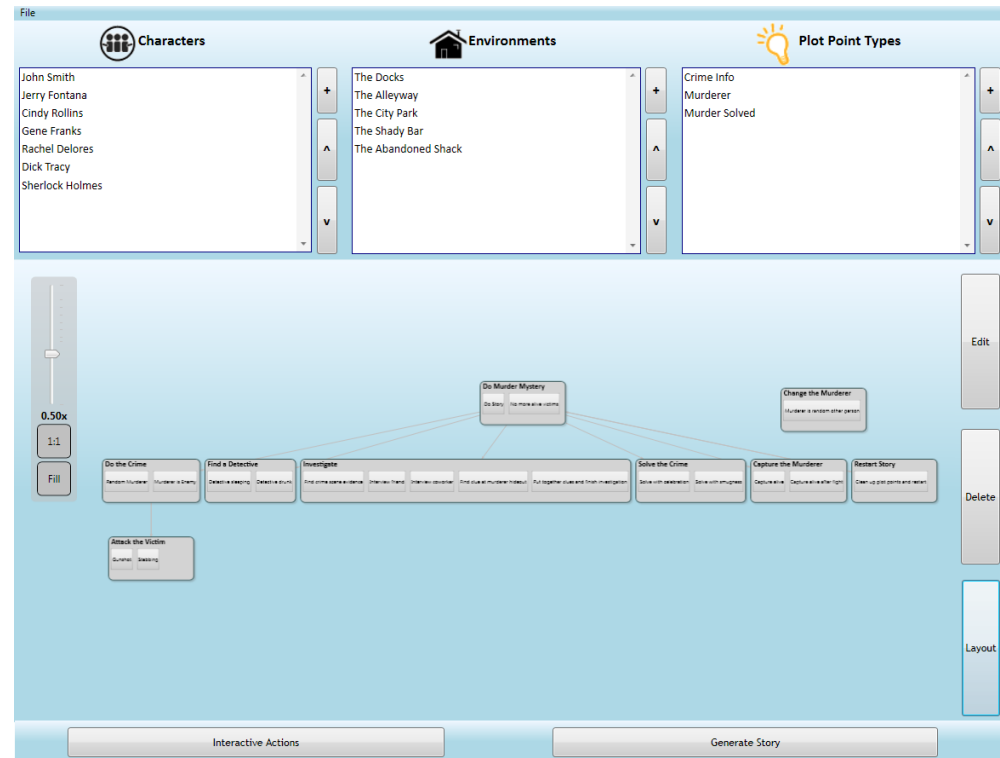
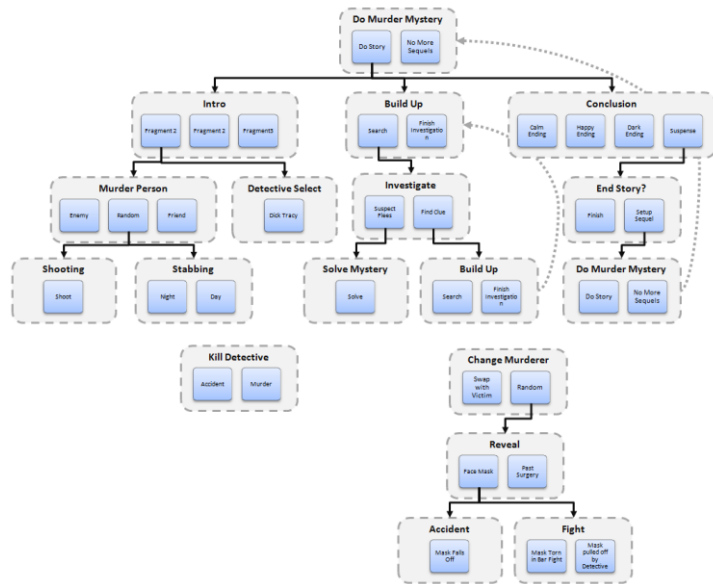
Wide Ruled Deliverable

- .wr2 file
- Story Quality + Story Structure
- Structural Requirements
 - Must run (must generate story)
 - Must use every feature at least once
 - Characters – traits and relationships, Environments – traits and relationships
 - Plot Points – traits
 - Author goal – parameters
 - Plot fragments
 - Preconditions: All types (Character, Environment, Plot Point)
 - Save a trait to a variable
 - Use variable within precondition
 - Actions
 - Print text (with inserted variables), subgoal, calculate value, edit character, edit environment, create plot point, edit plot point, delete plot point
 - Interactive Actions – at least one action that changes the story outcome

The future of Wide Ruled: **Story Canvas**

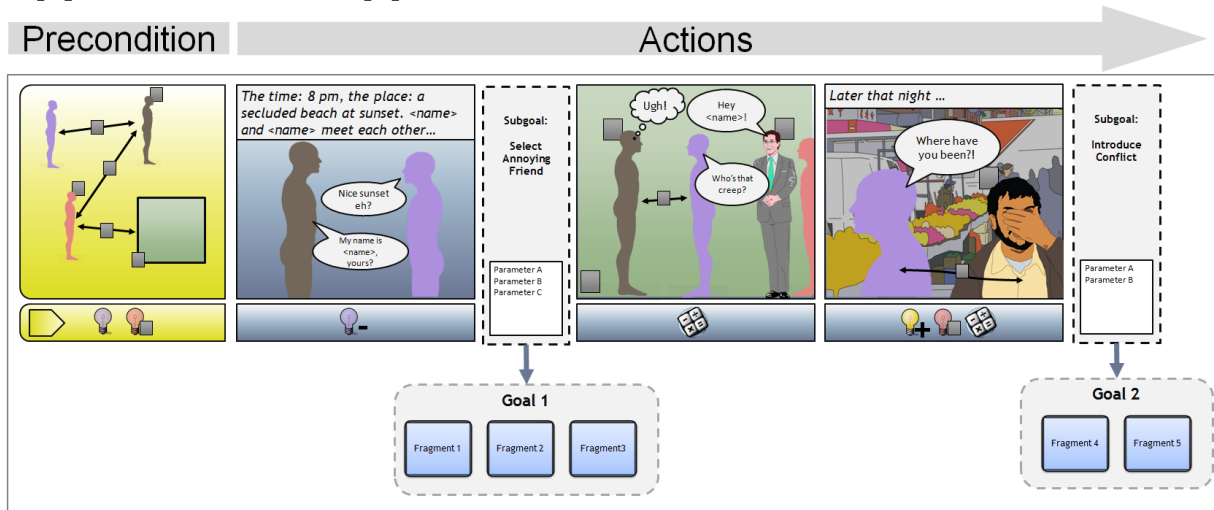
Story Canvas

- Visual, Storyboard-based authoring
- Same UNIVERSE-like story model, new graphical interface



Story Canvas

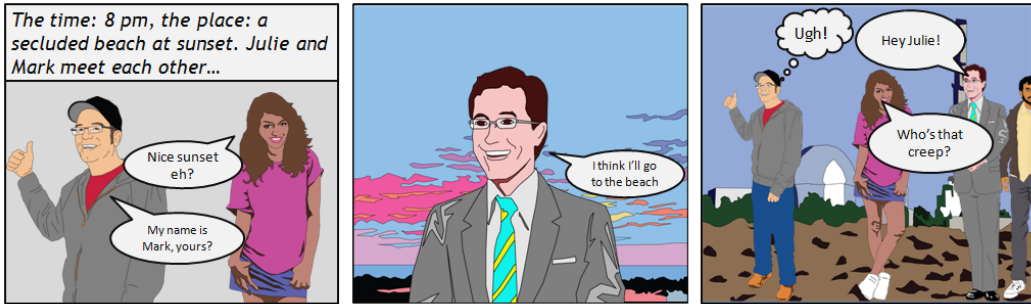
■ Editing Plot Fragments




Story Canvas

■ Reading generated stories

The time: 8 pm, the place: a secluded beach at sunset. Julie and Mark meet each other...



Later that night ...



Annoying friend finds out Secret lover dies Main character spills guts

<< >>

MurderMysteryInteractive



1.00x
1:1
Fill

Where have you been?

On a gloomy summer evening on the beach, Frank makes an important decision.

I'm gonna kill my wife!

Is Jerry your friend? How long have you known him?

Who? I have no idea!

Annoying friend finds out Secret lover dies Main character spills guts

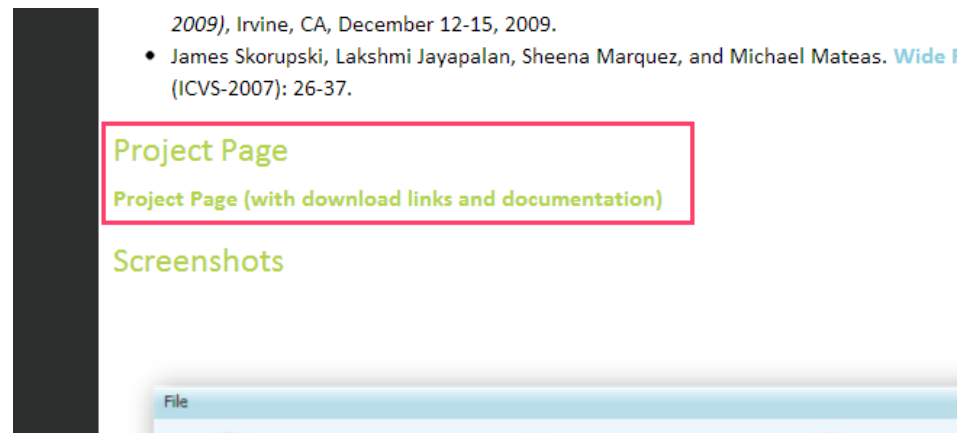
<< >>

Story Canvas

- Very early version for debugging/initial testing
 - fragile/buggy/etc
- Experimental usage for limited set of volunteers
- Email me (j Skorups@soe.ucsc.edu) if interested

Contact

- Website: [http://eis.ucsc.edu/Wide Ruled](http://eis.ucsc.edu/Wide_Ruled)
 - Click on “Project Page” link



- Support: Piazza + Email:
j Skorups@soe.ucsc.edu

Story Canvas

(Extra Slides!)

Story Canvas

■ Authoring Story Objects

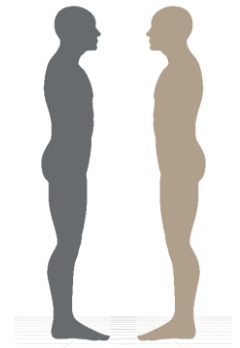
■ Static Characters and Environments

- WR2 char's/env's with:
- Avatar images
 - Provided and user assignable
 - Characters have a small finite set of poses



■ Dynamic (unbound during authoring)

- Anonymous colored icons

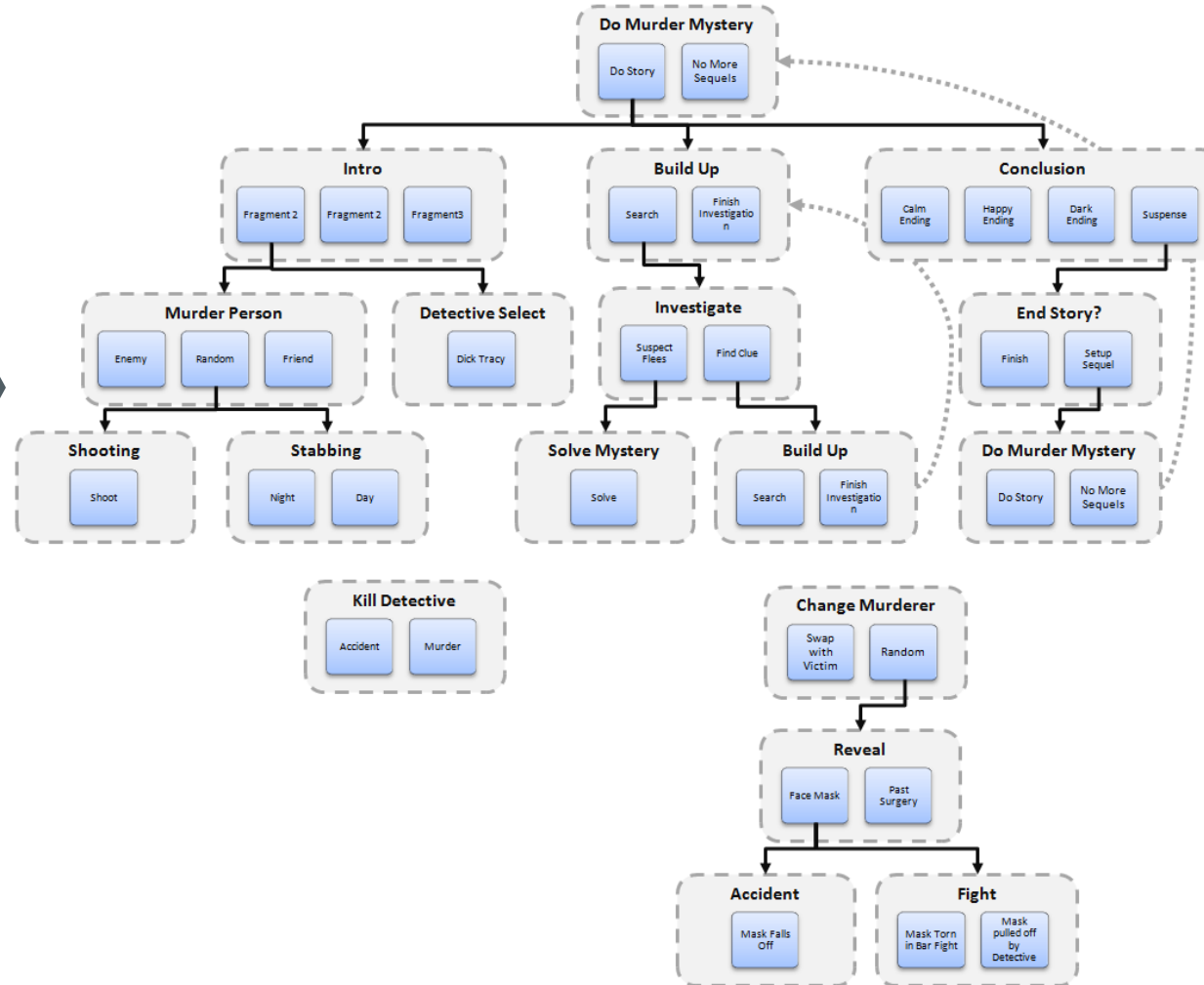


Story Canvas

■ Author goal and plot fragment hierarchy

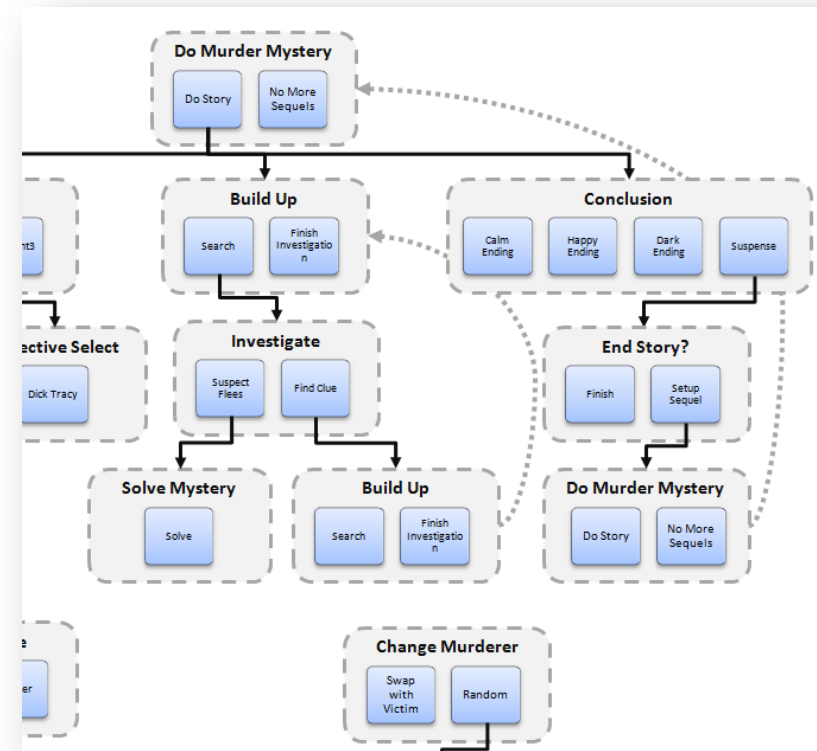
Goals and Plot Fragments

- ▲ Do Murder Mystery
 - ▷ Do Story
 - No more alive victims
- ▲ Do the Crime (Text Victim Name)
 - ▷ Random Murderer
 - ▷ Murderer is Enemy
- ▲ Attack the Victim (Text victim name)
 - Gunshot
 - Stabbing
- ▲ Find a Detective (Text Detective Name)
 - Detective sleeping
 - Detective drunk
- ▲ Investigate
 - ▷ Find crime scene evidence
 - ▷ Interview friend
 - ▷ Interview coworker
 - ▷ Find clue at murderer hideout
 - Put together clues and finish in
- ▲ Solve the Crime
 - Solve with celebration
 - Solve with smugness
- ▲ Capture the Murderer



Story Canvas

- Hierarchical List → Interactive Graph
- Goals encompass fragments
- Add/Edit/Delete Goals and Fragments
- Visualizes one or more hierarchies
- Visualizes recursion

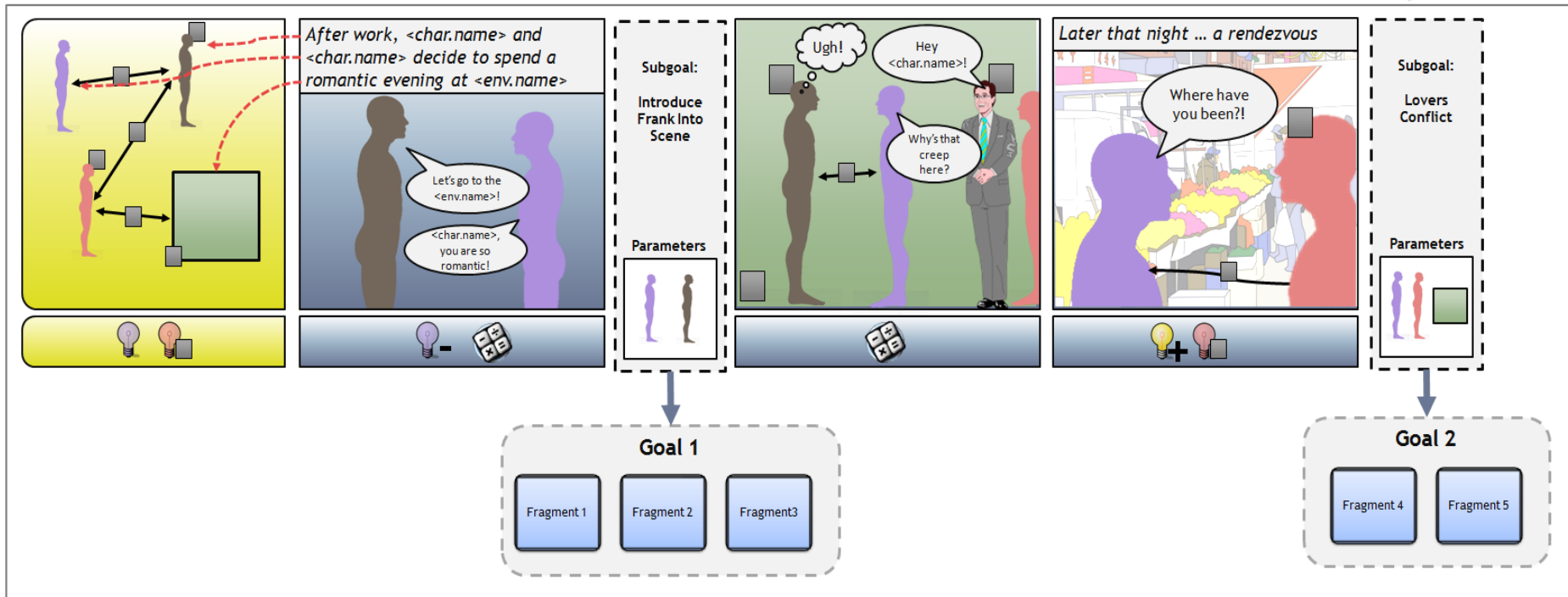


Story Canvas

- Visual precondition and story actions

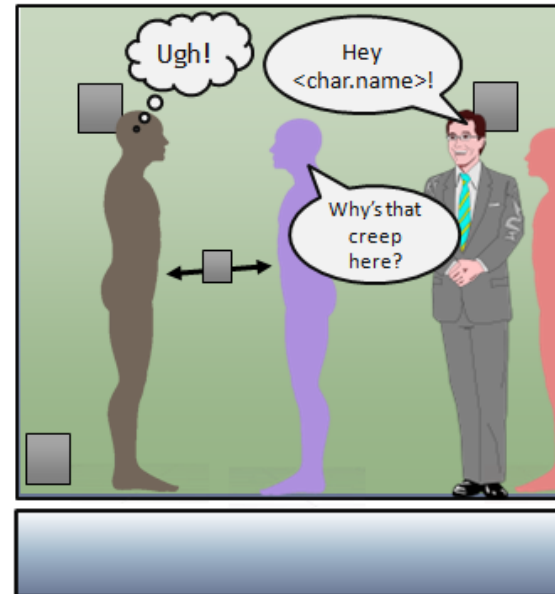
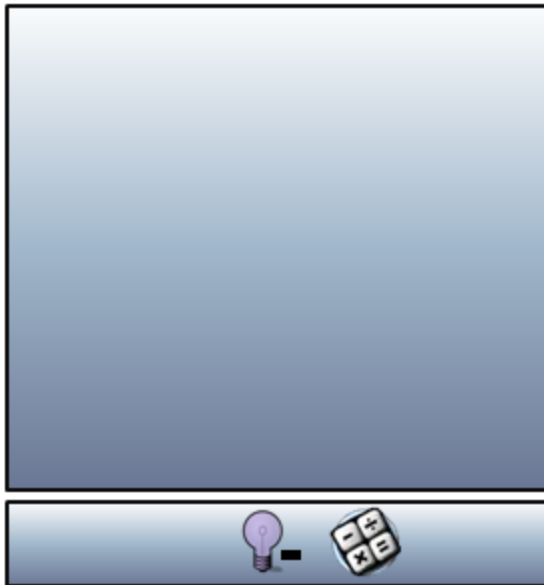
Precondition

Actions



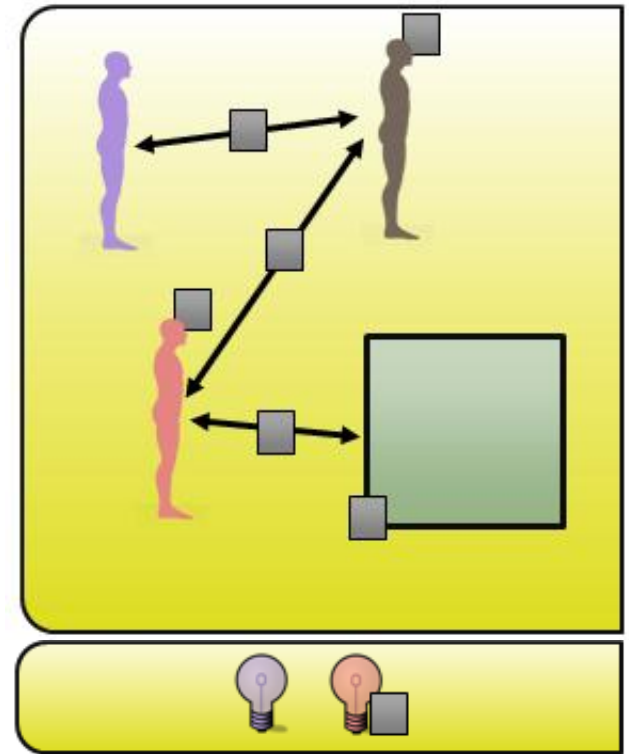
Story Canvas

- Visible and invisible pane pieces
 - Visible: Characters and Environments
 - Invisible: Plot Points and Calculations
 - Panes with no visible elements hidden in resulting story



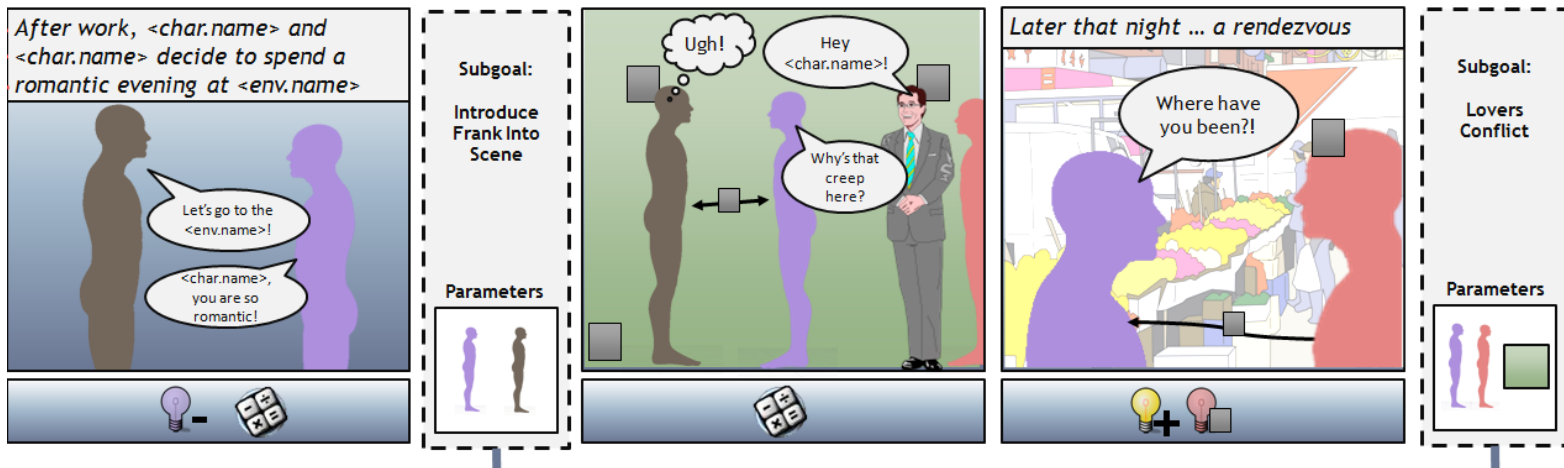
Story Canvas

- Precondition Pane
 - Two kinds of constraints
 - Inter-object (graph structure)
 - $\text{Char1.Age} > \text{Char2.Age}$
 - $\text{Char1.Enemy} == \text{Char2}$
 - Intra-object (floating box)
 - $\text{Age} < 24$
 - $\text{Name} == \text{"Bob"}$
 - $\text{Paranoid} == \text{false}$
 - Unbound characters / environments / plot points
 - Gray box = editable constraint lists
 - No saving and naming of temporary variables



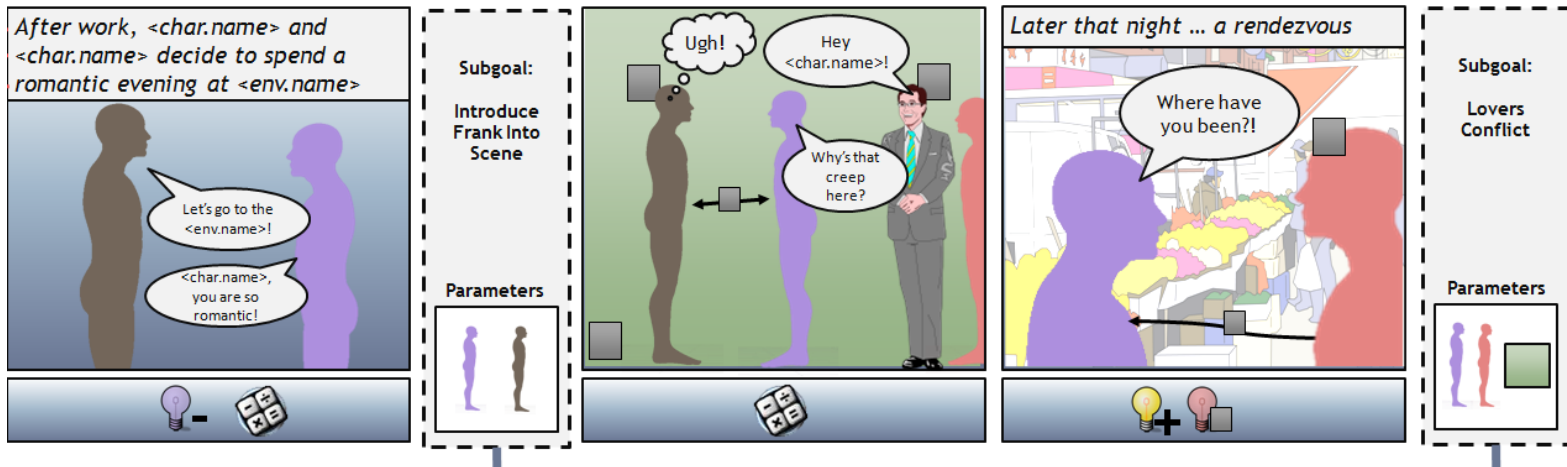
Story Actions

- Parameterized text output
 - Speech
 - Thought
 - Narration



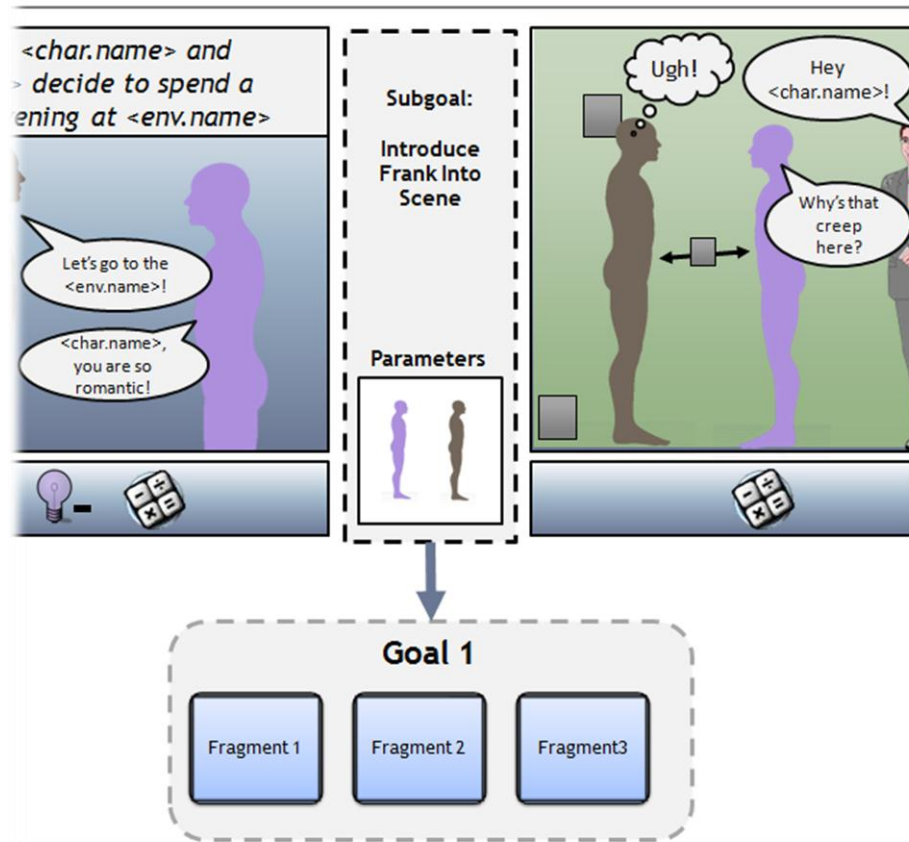
Story Actions

- Character/environment editing
 - Create relationship (graph structure)
 - Char1.Enemy = Char2
 - Char1.Friend = Jerry Smith (static)
 - Edit Trait/Relationship (floating box)
- Plot Point editing/creation/deletion
- Calculation of new values



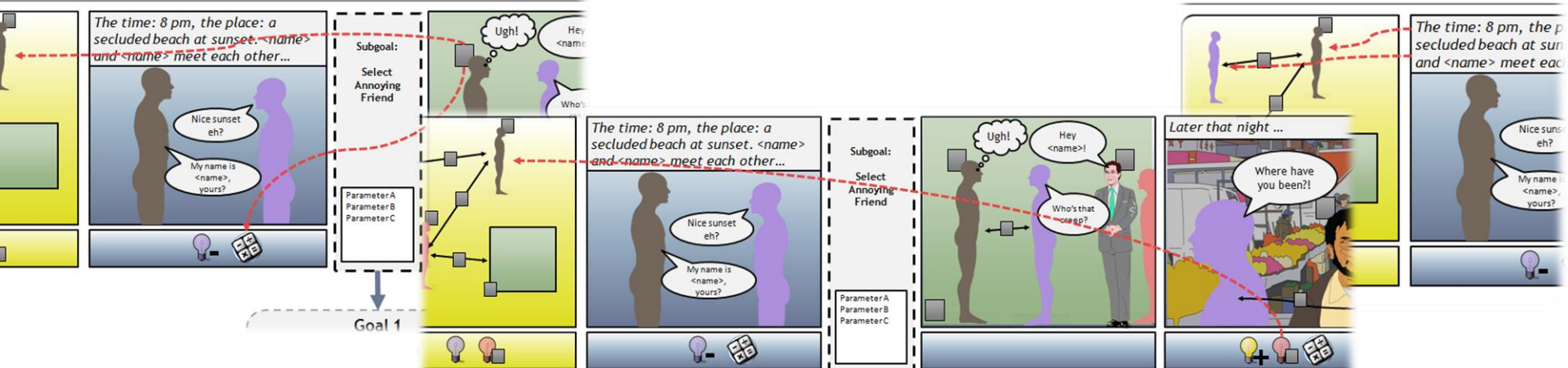
Story Actions

- Subgoaling with parameters



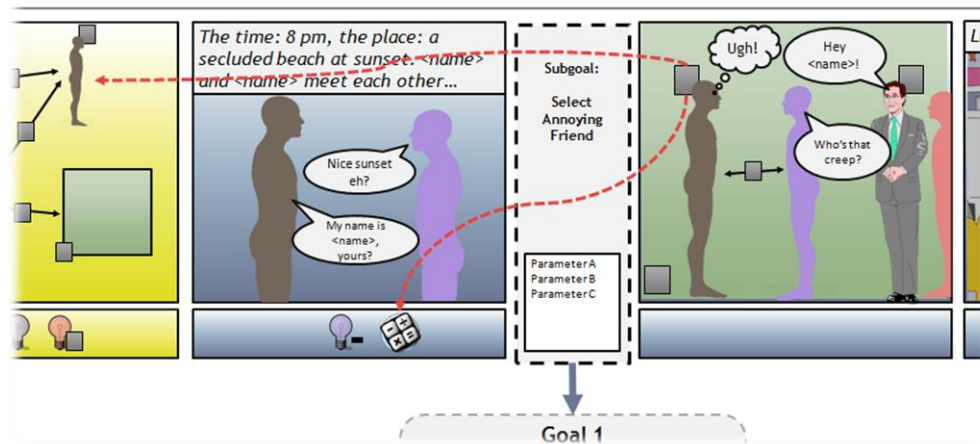
Story Action Variables

- Variable references
 - Author-customizable named matched object names
 - “Character1”, “TheBoyfriend”
 - References: Character1.Name, TheBoyfriend.Age
 - No individual variable names: char1name, char1age, etc
- All objects in preconditions and all of their contained information is always available
- Information dependencies visible when selected



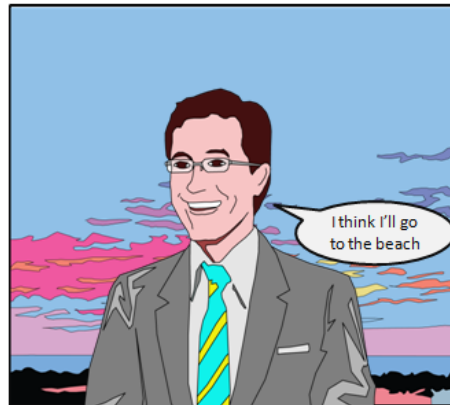
Variable Referencing

- Variable modification explicit in WR2 (ordered set of story actions)
- Visual domain means ambiguous ordering of assignments
- Solution: use panes as temporal guide – only reference information from previous panes



Reading

- Generated panes ground out dynamic story objects into static images
- Reader-controlled pace



Annoying friend finds out

Secret lover dies

Main character spills guts

<<

>>

Story Canvas

- Wide Ruled 2 → Story Canvas 1
 - Text-based → Comic book (**text + graphics**)
 - **List-based** story hierarchy → **visual graph-based** story hierarchy
 - Elimination of temporary variable management in preconditions
 - Elimination of variable names completely
 - **Global** interaction actions → **Contextual** interactive actions
 - Reader-controlled story pace
 - New features: static characters/environments, visual composition control