Getting Creative with Actions

David Thue

Abstract The goal of this chapter is to help refine and broaden how authors think about actions in the context of Interactive Digital Narrative. An action is generally something that a player can do to affect the progression of an interactive story, but the elements that actions can change are often limited to the state of the narrative world. Examples include moving the player's avatar through the narrative world because they typed "go north", or recording that the player has accepted a given quest. While a rich and diverse set of interactive stories has been made using only actions of this kind, we will see in this chapter how using other kinds of action can broaden the author's repertoire and enable new opportunities for player interaction.

1 Actions as Instruments of Change

To gain a deeper understanding of actions in Interactive Digital Narrative (IDN), it is useful to think carefully about how interactive stories *progress*; *i.e.*, what things change over time and what causes those things to change. To help with this sort of thinking, we will use Interactive Process Modelling (IPM), a framework for reasoning about interactive systems that works well for narratives and games [7, 8]. We introduce and explain the ideas of IPM gradually as this chapter proceeds.

Usually[1], only one thing can change as a story progresses: the state of the narrative world. Abstractly, a *narrative world state* describes the narrative world at a single moment during a story's progress. It does so using a collection of two elements: (i) the attributes of all of the world's entities at that moment, and (ii) a set of

David Thue
RISE Research Group
School of Information Technology, Carleton University, Ottawa, Canada, and
Department of Computer Science, Reykjavik University, Reykjavik, Iceland
e-mail: david.thue@carleton.ca

[1] We will see some alternatives in Section 2, but delay them until later for the sake of simplicity.

This is a preprint of the following chapter: David Thue, Getting Creative with Actions, published in "The Authoring Problem: Challenges in Supporting Authoring for Interactive Digital Narratives", edited by Charlie Hargood, David Millard, Alex Mitchell, and Ulrike Spierling, 2022, Springer Cham, reproduced with permission of Springer Cham. The final authenticated version is available online at: http://dx.doi.org/[DOI forthcoming, pending publication]".

information that summarizes the story's progress, up until that moment. The entities might consist of characters and objects, and attributes might include the position and orientation of every character and object, the goals and opinions of characters, the value or ownership of objects, and more. The information that describes the story's progress could include which quests the player has discovered, accepted, declined, or completed (if the IDN has quests). It might also include *flags* – unique markers which, when set, indicate that some particular happening has occurred in the story (*e.g.*, the player's first time meeting a particular character). The first element describes the contents of the narrative world, while the second element describes extra-world information that is useful for tracking player progress.

One of the goals of Interactive Process Modelling (IPM) is to model how states change as a result of player interaction. Modelling this process of change is useful because the model can help designers reason carefully about several important facets of an IDN. These include how each state change is mediated, what can change, who (if anyone) can change it, how they can change it, how they perceive it, and what they perceive.

1.1 The Interactive Process

According to IPM, a state changes as the result of one or more agents acting in a cyclic *interactive process*. An interactive process models how a particular *target object* (*e.g.*, a narrative world state) is perceived, reasoned about, and ultimately changed by one or more agents. The model represents each of these three steps of perception, reasoning, and change as an abstract function: an observation function, an action function, and a transition function, respectively (see Figure 1)[2]. We introduce each

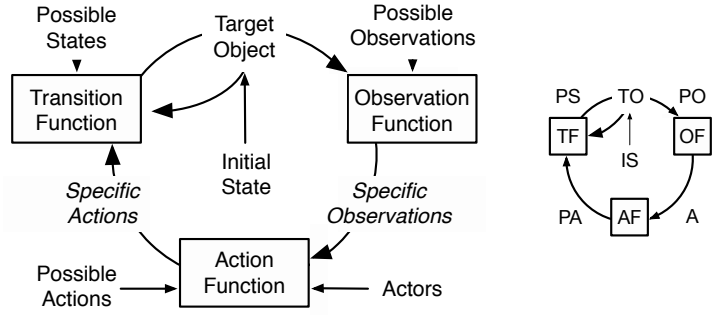

Fig. 1 Left: An interactive process. Boxes represent functions while labeled arrows represent data. Italics show data that only arises while the process executes. Right: A minified version of an interactive process.

[2] A function can be generally thought of as something that collects one or more inputs and uses them to produce one output.

functions briefly here, and follow with more detailed explanations in Sections 1.2 to 1.4:

- A *transition function* processes all agent actions as well as the current state of the target object, and produces a (potentially different) state, which becomes the new current state.
- An *observation function* processes the current state of the target object and produces an observation for each agent (*e.g.*, each player) that participates in the interactive process.
- An *action function* processes the observation that it receives and produces an action; this represents how an agent decides how to act based on what it observes.

In an interactive process, execution proceeds clockwise following the arrows in the figure, and time advances in a discrete way, by one *step* each time a new state of the target object is produced. Note that modelling time in discrete steps (instead of as a continuous value) is not restrictive in practice, because even player experiences of IDNs that *seem* continuous (*e.g.*, playing a modern computer role playing game) have discrete time steps. Such experiences only seem continuous because the time steps occur faster than most players will notice (*e.g.*, 60 times per second or more).

As Figure 1 shows, an interactive process has more elements than its three functions and target object. These remaining elements are defined as follows:

- A *set of actors* describes the set of agents that is able to participate in the interactive process by observing or attempting to change the target object – we say that an agent is an *actor* in the interactive process if it is in the set of actors. In very many IDNs, the set of actors contains exactly one member: the single player of the IDN.
- A *set of possible states* describes all of the possible configurations of the target object. For example, when the target object is the state of a narrative world, the set of possible states describes: (i) all possible ways of setting the attributes of the entities in that world as well as (ii) all of the possible records of story progress that might be recorded. Note that this set can easily be *very* large. As a simple example, consider a small narrative world with only 5 characters, each with 10 attributes (hair colour, favourite sandwich, *etc.*), where each attribute has its own set of 8 different values that it might take on (black, brown, *etc.*; tuna, peanut butter & jam, *etc.*). The number of possible states in this world is $8^{(5\times10)}$, which is a number so large it takes 46 digits to write down. Fortunately, an author never needs to write down all of the possible states one by one, and for this chapter, it is enough to understand that it can be convenient to consider all of the possible configurations of a target object abstractly, all at once, as a set of possible states.
- An *initial state* (which is one of the possible states) describes how the target object should be configured when execution of the process begins. When the target object is a narrative world state, the initial state describes the attributes that every entity should have at the beginning of the story, along with the starting configuration of any records of the story's progress.
- A *set of possible observations* describes all of the observations of the target object that an actor in the interactive process might receive. Like the set of possible states, the set of possible observations can easily be very large (*e.g.*, imagine all of the

ways of rendering a player's view of a complex 3D world). Nonetheless, it can be convenient to think of these possibilities all at once, as a single set of observations.
- A *set of possible actions* describes all of the actions that an actor in the interactive process is able to perform as they attempt to change the target object. When the target object is a narrative world state, the possible actions are determined by the author and might include "talk to character", "move", "pick up object", "use object", and more. We explain our notion of actions further in Section 1.1.1.

Following some clarifications of terminology, we explain each of the three functions of an Interactive Process in detail and highlight their relevance to player actions.

1.1.1 On the Meaning of "Action"

Before we proceed, it will be useful to explain what we mean by "action" with respect to some prior work. In 1976, van Dijk discussed the notion of action in the context of narrative as an intentional sequence of *doings* [9, 10]. For example, the action "open door" might consist of this sequence of doings: (reach for the door knob, grasp it, turn it, and pull back on it). In distinguishing between actions and doings, van Dijk's work helps us clarify the level of abstraction that we find appropriate for discussing actions. The abstraction level of doings is too low (too specific) to capture the actions that we wish to discuss, but considering each action to be a *sequence of doings* (which is a more abstract construct) suits our purposes well.

Although van Dijk considered interaction that might happen between the characters of a story, the notion of user or player interaction was not discussed. Later, however, Rafaeli defined "interactivity" as something that includes person-to-system communication, saying that what happens next must depend on what has already happened [5]. This dependence of future happenings on past happenings is modelled by IPM in the way that one or more states change as a result of agent interaction, as we will explain further in the sections that follow.

In 2007, Zagal *et al.* defined an action in the context of a game as being the manipulation of one or more entities in the game's world [11]. This notion of action maps well to a particular interactive process modelled using IPM: one whose target object is the state of the game's world. More specifically, if the interactive process shown in Figure 1 had a game's world state as its target object, then Zagal *et al.*'s notion of actions would describe the actions in Figure 1 well. For example, "open cellar door" could be an action that manipulates the entity "cellar door", changing part of the game's world state from "cellar door closed" to "cellar door open".

1.1.2 The System, Process, and Products of an IDN

In this chapter, we follow Koenitz's System, Process, Product (SPP) model [3] and use of two of its key concepts: an IDN system and an IDN process. An *IDN system* is a combination of art assets, program code, computing platform, and input devices that offers the potential for someone to have one of the (typically many) experiences that

might result from interacting with the system. An *IDN process* characterizes what occurs while a person interacts with an IDN system to have one of the experiences that it offers (each such experience is an *IDN product*). When we refer to "an IDN system's process", we mean the general IDN process that will occur for any person who interacts with that system. In this chapter, we use Interactive Process Modelling (IPM) to model what happens during an IDN system's process and relate it to both the program code of the IDN system and the IDN product that results. As we will see in the following sections, sometimes an IDN process can be usefully modelled as a *collection* of interactive processes, each with different target objects, whose transition functions and observation functions exist as code within the IDN system.

1.2 Transition Functions: How Actions Change States

What causes the narrative world state to change? Every such change is caused by a sequence of two steps:

1. the player performs an action (which might just be to wait), and
2. the IDN system responds by changing the state.

A common player action that changes the narrative world state is "move" – players use it to change the position of their avatar (if they have one), or of objects, within the narrative world. Another is the "choose dialogue option" action that players often perform during conversations with story characters. A record of their choice is often kept, either to unlock later dialogue or to trigger variation if the conversation is repeated. This information is stored in the narrative world state, and updating the record thus requires changing that state.

The way that the IDN system responds to each player action is governed by a collection of functions, each of which carries out some author-given instructions to determine what should happen next. For example, in a story made with Twine, the author must describe which node of content should appear next, for each available link that appears in any node. Put another way, they must describe how the narrative world state (including what node to display) should change, given the current world state and an available player action. When the Twine story is played, the player will perform actions (*e.g.*, by selecting links), and system must respond to each action in a way that carries out the instructions that the author provided.

Interactive Process Modelling (IPM) refers these response-governing functions as *transition functions*, because they help to transition some state from one configuration to another. Note that the player's ability to affect change though actions is thus *indirect* – it is mediated by the IDN system's transition functions (Figure 2). In an IDN system, the transition functions typically exist as program code that is *executed automatically*, meaning that their results are determined through digital computation. This is in contrast to Choose Your Own Adventure books (*e.g.*, *The Cave of Time* [4]), where the narrative world state (which is described by the current scene) only changes if

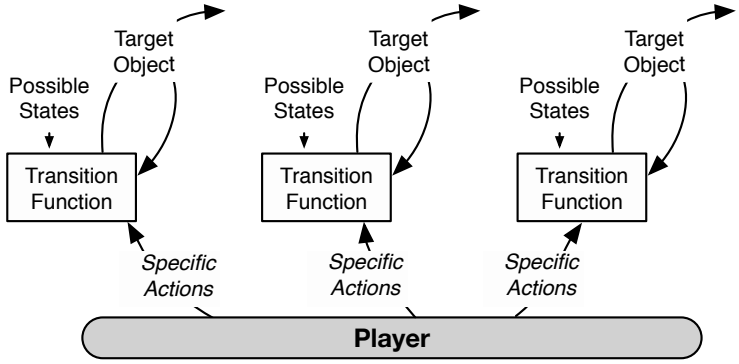

Fig. 2 An IDN system's different transition functions mediate how a player's actions are able to change different target objects within the system.

the book's transition function is executed *manually* (by flipping to the page indicated alongside a chosen action)[3].

1.3 Observation Functions: How States are Conveyed to Players

Thus far, we have described the narrative world state in the way that an omniscient oracle might know it, with clear and perfect knowledge of everything it contains. While this perspective can be useful for design-focused reasoning, it offers a poor representation of any player's likely perspective, because some aspects of the world state are usually obscured. For example, parts of the world state might be purposely hidden for dramatic effect (*e.g.*, concealing the identity of a criminal to build suspense), or until some condition is met (*e.g.*, a player might remain unaware of a quest until they find the character that offers it).

Understanding the player's perspective is important for understanding actions because players typically decide how to act based on what they observe. So, what determines a player's observations? Similarly to how an transition function determines what aspects of a target object should change as a result of a player's action, an IDN system's *observation functions* determine what parts of a state the player should get to observe (Figure 3).

For example, the observation functions of an IDN system made in Twine are responsible for rendering a page of content that displays the current node of the story and what player actions are available from that node. When the player performs one of the available actions, the authored transition functions are executed to update the narrative world state (*e.g.*, changing the current node to display), and the observation

[3] Thue demonstrated how IPM would model the (analog) system and process of a Choose Your Own Adventure book in a recent paper [7].

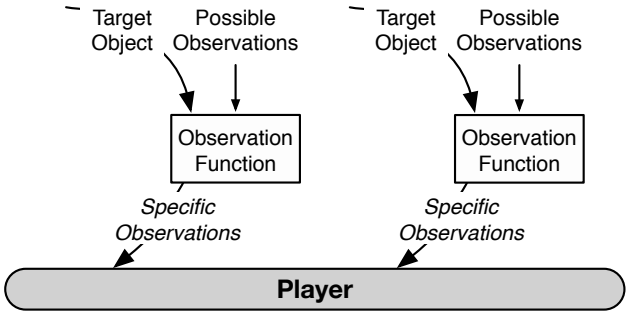

Fig. 3 An IDN system's observation functions mediate how the state of each target object can be observed by any player.

functions are then executed to produce new observations for the player, rendering the current node to the screen.

Like transition functions, the observation functions of an IDN system usually exist as program code that is executed automatically by the system.

1.4 Action Functions: How Observations lead to Actions

Having received one or more observations, a player must decide what to do next. IPM models this decision-making work using a third kind of function: the action function. An *action function* represents how a player considers (or ignores) their observations and then chooses an action to perform (Figure 4). Unlike how transition functions and observation functions are executed by the IDN system, action functions are executed by players (*i.e.*, each player does the work to determine their desired action). Continuing our example in Twine, an action function would model the player's reasoning about what they observe (*e.g.*, the text, links, and other content displayed on screen) as well as how they should behave as a result (*e.g.*, they might choose to open an available door).

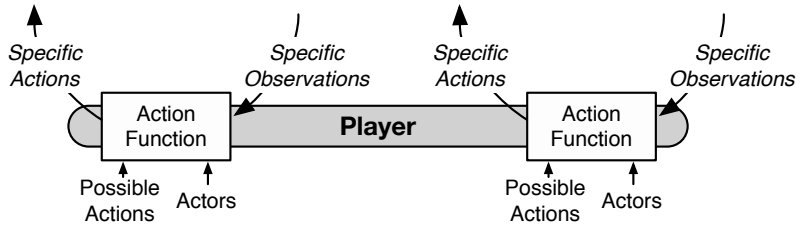

Fig. 4 An IDN system's action functions represent how a player uses their observations to decide what actions to perform.

2 Understanding Kinds of Action

In this chapter, we will distinguish between different kinds of action based on how they aim to change different parts of an IDN system. An action that seeks to change the narrative world state is thus one kind of action, but what might other kinds be? Put differently, what of an IDN system might be changed, other than the narrative world state? Reed offered some examples in his 2017 dissertation [6], two of which we highlight here.

In a mode of interactive narrative that Reed calls "sculptural fiction", players can change the ways in which the state of the narrative world might progress. Reed explains that instead of performing actions to traverse a pre-connected web of narrative content, players of sculptural fiction must decide how to connect together a given set of narrative content, to define how the state of the narrative world could potentially progress. In Interactive Process Modelling, the way in which the state of any target object progresses is modelled by the transition function of an interactive process (recall Figure 1). If we were to create an interactive process whose target object was the state of the narrative world, then sculptural fiction would allow players to change the transition function of that process.

Reed also discussed a second mode of interactive narrative, which he called "collaborative storygames"; *Dungeons & Dragons* [1] is an example of such a game[4]. Reed found that players of such games often engage in an activity he called *generation*. Generation refers to the creation of new content, such as inventing a new character and adding them into the narrative world. An action that generates new content is different from an action that changes the narrative world state, because generation redefines what states the world could possibly be in. In IPM, the possible states of any target object are described by a set of possible states. If we were to create an interactive process that targets the state of a narrative world, then generation would allow players to change that process's set of possible states.

2.1 Different Kinds of Action have Different Target Objects

Interactive Process Modelling offers a way to model how a transition function or a set of possible states can be changed as a result of player actions. To do so, one sets the desired target for change (*e.g.*, the transition function of some interactive process) as the target object of an *additional* interactive process. This is possible because IPM's notion of a target object is quite broad. In particular, the term "object" should be understood both abstractly and flexibly – a target object could potentially be any part of an IDN system, so long as that part can be changed as a result of one or more agents' actions. The primary purpose of an interactive process is to model how its target object can be changed by the actions of one or more agents.

[4] Although *Dungeons & Dragons* and similar games are not digital, they nonetheless served (and continue to serve) as inspiration for an entire genre of IDN systems: Computer Role-Playing Games.

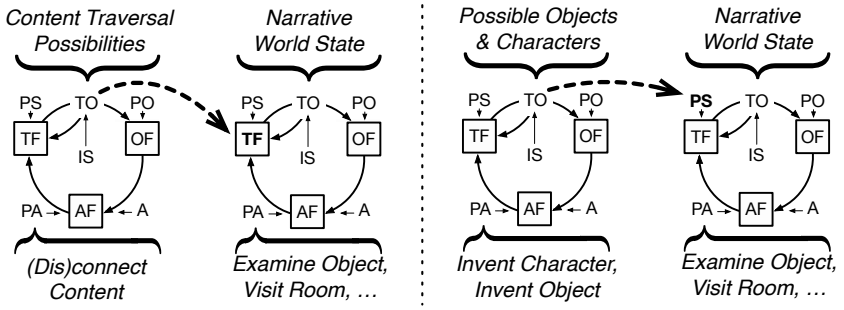

Fig. 5 Left: a partial interactive process model involving Reed's sculptural fiction [6]. Right: a partial interactive process involving Reed's idea of content generation. Both: The target object of each process is labelled across the top, and some possible actions in each process are labelled across the bottom. The thick dashed arrow shows a *target object link* (to be explained in Section 2.2).

As an example, Figure 5 shows partial interactive process models for both sculptural fiction (Left) and generation in a collaborative storygame (Right), continuing from our discussion of Reed's work above. Both models have one interactive process whose target object is the narrative world state, plus a second interactive process whose target object is an element of the first process. Taken together, the two models offer three different kinds of action to players: one kind that seeks to change the narrative world state, a second kind that seeks to change the transition function that governs the narrative world state, and a third kind that seeks to change the set of possible world states. In the following section, we will explore a general method for building such models and identifying different kinds of action.

As we suggested above, one way to distinguish between actions is to group them based on the element of an IDN system that they seek to change. While Reed identified a few changeable elements that are different from a narrative world state [6], he did not offer any general method to identify other potential elements. Interactive Process Modelling offers such a method, by modelling an IDN system's process and relating it to both the system's program code and the IDN's product [7].

2.2 Building an Interactive Process Model

The *interactive process model* of an IDN system is a set of one or more interactive processes, each of which represents how some element of the system or its process can change. The model begins empty (containing no interactive processes) and grows incrementally across the following sequence of steps.

2.2.1 Step 1: Identify Agents and Initial Objects

The first step in building such a model is to identify two things: (i) which agents should be considered as potential actors, and (ii) an initial set of objects that they can observe or change. Across very many IDN systems, the answers to these questions are the same: (i) a single player, and (ii) they can observe and change the narrative world state. As a more complex example, consider Reed's analysis of sculptural fiction [6] through the lens of IPM. We identify that (i) a single player should be considered an actor, and (ii) two objects can be observed and changed: the narrative world state, and the transition function of an interactive process that targets the narrative world state.

2.2.2 Step 2: Build the Initial Model

Once one has identified the agents to consider and an initial set of n objects to be observed or changed ($n \geq 1$), the next step is to create and add n interactive processes to the model. Since each identified object can be observed and/or changed in some particular way, the goal is to use a unique interactive process to model how that change happens. We do this by setting the target object of each interactive process to one of the identified objects, until every identified object is the target of exactly one interactive process. Applying this step is how we arrived at the partial models shown in Figure 5. For sculptural fiction [6], we created two interactive processes: one with the narrative world state as its target object, and a second process with the transition function of the first process as its target object (Figure 5: Left). For a collaborative storygame where players can create new narrative world states (which Reed called "generation" [6]), we also created two interactive processes: one process to target the narrative world state, and one to target the first process's set of possible states (Figure 5: Right). When the target object of one interactive process is an element of another, we say that the target object of the first process is connected to the element of the second process by a *target object link*. We show such links graphically as dashed arrows in our figures.

2.2.3 Step 3: (Potentially) Expand the Model

Once we have an interactive process for each of the target objects that were identified in the previous step, the next step is to check whether other observable and/or changeable objects might have been missed. One way to do this is to recursively examine the existing model's elements, asking the question "can any agent observe or change this element?" for each one. By "element" we mean any function or set that is part of the current model's interactive processes. Whenever the answer for any element is yes, a new interactive process is added to the model with that element as its target object. The examination is recursive because for each new interactive process that gets added along the way, one must eventually examine all of *its* elements and

consider whether or not each of them can be observed or changed. Eventually, the answers for all remaining elements will be "no", and the recursive analysis will end.

Note that although IPM can reveal many potential targets for change, it does not claim to fully cover every possible element of an IDN system and its process. As a result, the recursive analysis might not find some elements that were missed in the first step of modelling. Such an element can nevertheless be added to the model as soon as it is discovered, by creating a new interactive process with the element as its target object and performing a recursive analysis of the new process's elements.

2.2.4 Step 4: Assign Actors

Given *m* interactive processes with their target objects assigned to different, changeable elements of the IDN system or its process, the next step is to add one or more agents to each interactive process's set of actors. We do so by asking a simple question of every interactive process in the model, considering each of the agents that were identified in the first step above: can this agent observe or change this process's target object? When the answer is "yes", the agent is added to that process's set of actors. The sets of actors will be complete when every pair of interactive process and agent has been considered.

Each addition or omission of any agent to these sets will ultimately describe which target objects that agent is or isn't able to observe or try to change. For example, in some games of Dungeons & Dragons, observing and changing the set of possible world states is only be available to a player who has a special role (*e.g.*, the "Game Master"); this means that only the Game Master would be added to the set of actors for the interactive process that targets the set of possible world states.

2.3 Influence Paths: How Actions affect Narrative Experiences

A completed interactive process model will have *m* interactive processes, and each of these will allow its actors to perform a different kind of action than any other process in the model allows. Distinguishing between different kinds of action in this way is useful because they all affect a player's narrative experience (the *product* of the SPP model [3]) through a different path of mediating elements in the IDN system. We call these paths *influence paths* [8], because they can be traced across an interactive process model to show how an agent's action in one interactive process will ultimately influence their narrative experience – sometimes in a variety of different ways.

As an example, consider an IDN that uses sculptural fiction, where a player wishes to make it possible to traverse between two nodes of narrative content (see Figure 6). To accomplish their goal, the player can perform a "connect" action in interactive process B. When performed, this action has two effects on the player's experience, each through different influence paths.

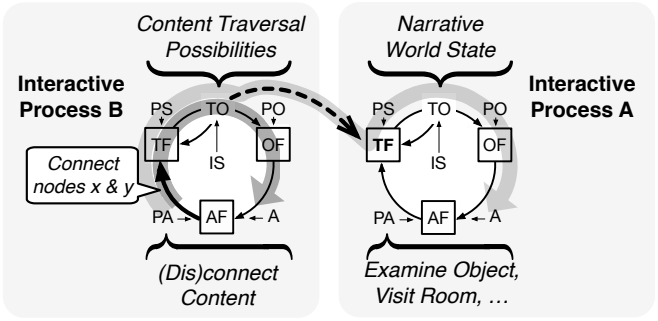

Fig. 6 A demonstration of how influence paths can be traced within and across the processes of an interactive process model. Two influence paths are shown – one in dark grey (fully contained within interactive process B), and one in light grey (which crosses from B to A). The example comes from an IDN that uses Reed's sculptural fiction [6].

One path remains local to the interactive process B (shaded dark grey in the figure), with the effects of the player's action being mediated by two elements of the IDN system:

1. B's transition function mediates the effect of the player's action on the state of B's target object: assuming that it allows the requested connection to be made, it changes the state of the target object by adding a connection between the indicated nodes of content.
2. B's observation function mediates the effect of this state change on the player's experience: it produces an observation that informs the player that the desired connection between content has been made.

The second influence path (shaded light grey in the figure) crosses from B to A via B's target object link to A. This link exists because B's target object is A's transition function. When the player performs the "connect" action in B, the effects of that action are mediated along this second influence path as follows:

1. B's transition function mediates the effect of the player's action on the state of B's target object: assuming that it allows the requested connection to be made, it changes the state of the target object by adding a connection between the indicated nodes of content (this step is the same as above).
2. Crossing the target object link from B to A, A's transition function is now different than it was before (because B's transition function just changed it). From this point in time onward, A's transition function will mediate the effect of any actions performed in A differently than it did before. As a result, the action that our player performed in B will have a potentially lasting effect on how A's target object (the narrative world state) changes.
3. As the narrative world state changes according to A's transition function, the effects of these changes will be mediated by A's observation function: it will

produce observations that (potentially[5]) inform the player about what changes are happening in the world state.

By examining the influence paths of an interactive process model, authors can better understand how different kinds of action might affect not only the operation of various parts of an IDN system, but also the various observations that a player could receive as they interact within that system's process.

2.3.1 Finding a Model's Influence Paths

To enumerate a given model's influence paths, one begins at the action function of each interactive process in the model. For each action function, a new influence path extends clockwise to the first to the transition function of the same process, and then to its target object. If the target object is linked to an element of another interactive process via a target object link, then the influence path is duplicated. One copy carries on within same the process, first to the observation function and then to the action function from which it started. The other copy crosses the target object link to the element of the interactive process at the other end. It then carries along clockwise across the elements of the process, duplicating as needed to both carry on within the same process as well as cross any target object link that it encounters. Every influence path ends as soon as it reaches any of the model's action functions.

3 Exploring the Landscape of Different Kinds of Action

Section 2 explained how Interactive Process Modelling can be used in an analytical mode, to gain an understanding of what different kinds of action are made available by an IDN system and how they relate to players' experiences therein. In this section, we discuss how IPM can be used in an *exploratory* mode, toward discovering and making sense of kinds of action that go beyond what IDN systems typically offer.

To explore different kinds of action using IPM, we modify Step 3 of the modelling procedure (Section 2.2) as follows: instead of asking "can any agent observe or change this element?", we ask "should any agent be able to observe or change this element?" Whenever the answer is "yes", we create a new interactive process with the identified element as its target object. Unlike in the analytical mode, where the elements of each new interactive process represent existing parts of an IDN system, the elements of new processes that are created in the exploratory mode might not yet exist. For example, consider answering "yes" to the question "should any agent be able to observe or change the set of possible actions that players can use to modify the narrative world state?". While our question is about an existing element of an IDN system (the set of possible actions highlighted in Figure 7, our "yes" answer prompts the creation of a new interactive process (C, in the figure) whose elements

[5] Recall that some parts of the world state might be hidden from the player's view.

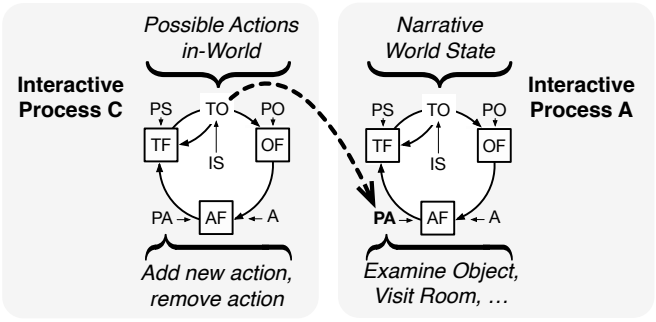

Fig. 7 When we answer "yes" to "should any agent be able to observe or change the set of possible actions that players can use to modify the narrative world state?", we create Interactive Process C to model how Interactive Process A's set of possible actions (**PA**) can change.

do not yet exist. By building the Interactive Process Model, the author is prompted to consider several important questions:

- Which agents should be able to observe or change A's set of possible actions? (C's set of actors)
- How should observations of A's set of possible actions be produced for each of C's actors? (C's observation function)
- What actions (in C) should C's actors be able to perform? (C's set of possible actions)
- How should A's set of possible actions be allowed to change via C's transition function? (C's transition function)
- How should A's set of possible actions be when the story starts? (C's initial state)

After creating a new interactive process in response to each "yes" answer to our modified question, we continue with the recursive part of Step 3, asking the same question of every element of every new interactive process, until no new elements remain to ask about.

An author who answers "yes" many times during this recursive analysis will be prompted to consider a wide variety of potential kinds of action - each of which seeks to change a different (and perhaps not-yet-existent) element of an IDN system. Figure 8 shows one potential interactive process model that could result from such an exploration.

Summary

Interactive Process Modelling (IPM) allows an author to distinguish between different kinds of action in a particular way: based on which part of an IDN system they aim to change. By modelling an existing IDN system and examining its influence paths, authors can better understand how each of the different kinds of action that players can

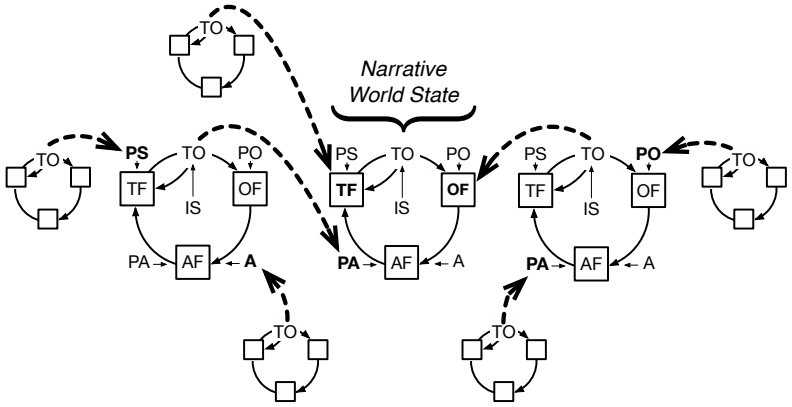

Fig. 8 An example of an exploratory interactive process model that could be generated by answering "yes" to the question "should any agent be able to observe or change this element?" for each of the elements shown in bold (and "no" for all other elements). Each of the eight interactive processes in the model allows a different kind of action.

perform might ultimately affect their experience. Distinguishing between different kinds of action in this way is useful for innovation; the vast majority of IDN systems only allow the state of their narrative world to change, and they restrict themselves to offering only one kind of action as a result. By modelling an IDN system using IPM as the system is being designed, authors can explore a broad range of potential targets for player-affected change, and be guided through key decisions about how that change should happen. Actions in IDN are capable of more than what they are commonly used to do, and the methods presented in this chapter aim to realize that potential.

References

1. Gygax, G., Arneson, D.: Dungeons & Dragons. Tactical Studies Rules, Inc. (1974)
2. Kleinman, E., Caro, K., Zhu, J.: From immersion to metagaming: Understanding rewind mechanics in interactive storytelling. Entertainment Computing **33**, 100322 (2020). DOI 10.1016/j.entcom.2019.100322
3. Koenitz, H.: Towards a theoretical framework for interactive digital narrative. In: R. Aylett, M.Y. Lim, S. Louchart, P. Petta, M. Riedl (eds.) Interactive Storytelling, pp. 176–185. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
4. Packard, E.: Choose Your Own Adventure: The Cave of Time. Crossroads Press (1979)
5. Rafaeli, S.: Interactivity: From new media to communication. Advancing Communication Science: Merging Mass and Interpersonal Process – Sage Annual Reviews of Communication Research **16**, 110–134 (1988)
6. Reed, A.: Changeful tales: Design-driven approaches toward more expressive storygames. Ph.D. thesis, UC Santa Cruz (2017)
7. Thue, D.: What might an action do? Toward a grounded view of actions in interactive storytelling. In: Proceedings of the 13th International Conference on Interac-

tive Digital Storytelling (ICIDS'20), LNCS Vol. 12497, pp. 212–220. Springer, Cham, https://rise.csit.carleton.ca/pubs/Thue_ICIDS_2020.pdf (2020)
8. Thue, D.: Supporting the design of flexible game systems. In: Extended Abstracts of the 2021 Annual Symposium on Computer-Human Interaction in Play, CHI PLAY '21, pp. 216–221. Association for Computing Machinery, New York, NY, USA (2021). DOI 10.1145/3450337.3484223
9. van Dijk, T.A.: Action, action description, and narrative. New Literary History **6**(2), 273–294 (1975). URL http://www.jstor.org/stable/468420
10. van Dijk, T.A.: Philosophy of action and theory of narrative. Poetics **5**(4), 287–338 (1976). DOI https://doi.org/10.1016/0304-422X(76)90014-0. URL https://www.sciencedirect.com/science/article/pii/0304422X76900140
11. Zagal, J.P., Mateas, M., Fernández-Vara, C., Hochhalter, B., Lichti, N.: Towards an ontological language for game analysis. In: DiGRA '05 - Proceedings of the 2005 DiGRA International Conference: Changing Views: Worlds in Play (2005). URL http://www.digra.org/wp-content/uploads/digital-library/06276.09313.pdf