Advances in Cognitive Systems 8 (2020) X-Y Submitted 5/2020; published 8/2020

The Visual Narrative Engine:
A Computational Model of the Visual Narrative Parallel Architecture

Chris Martens MARTENS@CSC.NCSU.EDU
Department of Computer Science
North Carolina State University, Raleigh, NC 27695 USA

Rogelio E. Cardona-Rivera ROGELIO@CS.UTAH.EDU
School of Computing and the Entertainment Arts and Engineering Program
University of Utah, Salt Lake City, UT 84112 USA

Neil Cohn NEILCOHN@VISUALLANGUAGELAB.COM
Tilburg Center for Cognition and Communication
Tilburg University, 5000 LE Tilburg, The Netherlands

Abstract

This paper introduces the first computational model of a proposed set of cognitive structures and processes involved in interpreting visual narratives (specifically, comics). The study of cognitive processes that occur when readers interpret comics has been theorized and experimentally validated to include spatial semantics—a mental model of the physical environment depicted in each image—and event structure semantics—a hierarchy of narrative events that take place over time. These semantic models are interconnected, developing in parallel as readers interpret a panel sequence. Our computational model aims to bring clarity to the cognitive theory of visual narrative sensemaking and poses new questions for the cognitive science community. Towards this end, we present a prototype computational model in which event structures are represented as hierarchical plans and spatial structures are represented as relational scene graphs. Domain knowledge about narrative events and how they relate to underlying scene structure is encoded in a standard Hierarchical Task Network (HTN) planning domain representation. Using a standard implementation of HTN planning, we demonstrate how to search the space of possible HTN solutions for ones that match with comic panel sequences.

1. Introduction

Visual narrative processing is a fundamental part of human cognition, with drawn image sequences "appearing on cave paintings, wall carvings, ancient pottery [and] extend[ing] across human cultures and time periods, making them a fundamental and universal part of human expression" (Cohn & Magliano, 2019, p. 2). This prevalence makes their study within cognitive science paramount. Recently, Cohn (2019b) developed *PINS* (Parallel Interfacing Narrative-Semantics), a *conceptual model* (Sun, 2008) of visual narrative reasoning that frames it as an interplay between syntactic and semantic processing. The knowledge representation of the PINS reasoning model is the Visual

Narrative Grammar (VNG, Cohn, 2019a), another conceptual model that posits the elements across syntax and semantics that interact in PINS, as well as the canonical forms of those elements. Taken together, the Visual Narrative Parallel Architecture (VNPA=VNG+PINS) links meaning, modality, and grammar to account for how sequences of *visual* "phonological" symbols (*e.g.* lines, shapes) are interpreted across different levels of understanding, such as communicative (discursive) structure, the spatial layout and referents of scenes being depicted, and their underlying event models.

However, despite progress, we still lack a computationally precise understanding of how this visual narrative processing works. The VNPA is a model presented at a level of abstraction that does not resolve ambiguities that arise from being computationally precise (*cf.* Marsella & Gratch, 2009), potentially resulting in a surfeit of more-specific models that describe sensemaking phenomena equally well (*cf.* McNamara & Magliano, 2009) with no way to disambiguate between them. This precludes a principled understanding.

In this paper, we develop the *first* computational model of the Visual Narrative Parallel Architecture, called the Visual Narrative Engine (VNE). The VNE operationalizes the VNPA by starting from its VNG representational theory and imputing a computational procedure to the operations the PINS uses to describe how the (semantic) event structure is mentally built-up from sequences of (phonological) images. As a first-step toward defining *all* the described operations in the VNPA, we assume that the syntactic structure and semantic knowledge is built into the form and content of the event structure. We focus on PINS' Semantic Prediction operations, *i.e.* those governing aspects of event inferencing (Graesser et al., 1994), through an algorithm that accepts a representation of a comic (sequence of visual narrative frames) as input, and produces a hierarchical event model as output.

Our contribution is two-fold. First, we demonstrate an existence-proof that we *can* readily describe a subset of the posited mental operations described in the VNPA in mechanical terms. Second, we identify gaps in the Theory of Visual Language necessary to answer for us to achieve a fully computational model; these gaps become future questions to answer both in cognitive science and cognitive systems research.

2. Related Work: Computational Narrative Sensemaking

The computational study of narrative is rooted within artificial intelligence (AI) and focuses on developing executable systems that model narrative *generation* (Gervás, 2009) and *sensemaking* (Mueller, 2013). Laubrock & Dunst (2019) reviewed state-of-the-art computational models as applied to the sensemaking of visual structure, narrative and text structure, and reading within comics. Most techniques are heavily grounded in document analysis, and prior work has focused on visual structure processing, which includes panel segmentation, page layout, caption/balloon segmentation, character/object detection, and stylometry (Laubrock & Dunst, 2019, Table 2). The authors note that "higher-level conceptual structure is still a relatively neglected topic in computational comic analysis, as are most sequential aspects, especially *narrative and event structure*" (Laubrock & Dunst, 2019, p. 4, emphasis added). Of their surveyed work, that by Iyyer et al. (2017) is most relevant: they developed a recurrent neural network that, given a pair of adjacent comic panels, predicts what subsequent text or image (from 3 candidates) might follow, a form of cloze task (Saraceni, 2016).

Our work is distinct from theirs in several ways: our model is based on symbolic representation and manipulation, we attempt to model the ("relatively neglected") cognitive process at play during comic comprehension as opposed to their aim of succeeding at a cloze-task, and we assume that relevant knowledge has been extracted from the comic itself as opposed to gleaned from a (computer-)visual scan of the comic's surface code.

We target a narrative sensemaking *cognitive system* (Langley, 2012), which simulates in software the posited cognitive processes at play during the consumption of visual narrative in a manner that is not *functionalist* (Searle, 1980). That is, the VNE's construction is *cognitively-grounded*, *i.e.* constructed under human cognitive constraints empirically identified within cognitive neuroscience (which gave rise to the VNG and PINS models). Most directly-related work is functionalist (see Mueller, 2013, for a review), but there are two notable exceptions. First is the work by Martens & Cardona-Rivera (2016): while focused on comic *generation*, their work appeals to theories of visual narrative sensemakingas a way to guide a generative algorithm. Second is the work by Cardona-Rivera et al. (2016), which models a reader's cognitive state *after* reading a narrative on the basis of work within story psychology that frames readers as problem-solvers (Gerrig & Bernardo, 1994). They use an automated planning knowledge representation (Ghallab et al., 2004), but do not explicitly target anything visual, nor do they model the process of sensemaking, which is our goal.

3. The Visual Narrative Engine

In this paper, we take the first step toward defining the Visual Narrative Engine by (a) defining computational analogues to the conceptual structures defined in the VNPA, and (b) defining one computational procedure that simulates the process of mentally constructing the event models on the basis of the phonological structure of sequential images. Our aim is to develop the VNE as a *cognitively-plausible structural model* (Lieto & Radicioni, 2016) of human visual narrative processing psychology by respecting the representational (VNG) and reasoning (PINS) constraints identified within the VNPA.

3.1 Conceptual Basis: The Visual Narrative Parallel Architecture

The VNPA is the conceptual model that we seek to operationalize in this paper. Doing so involves operationalizing its knowledge representation—the VNG—and its reasoning mechanisms—the PINS model. Next, we discuss each of these in turn, using a segment from the comic *Savage Dragon* (Larsen, 1993) as a running example (see Figure 1).

3.1.1 Knowledge Representation: The Visual Narrative Grammar

The VNPA's representational model, the **Visual Narrative Grammar** (VNG), is a theory that explains the ways in which sequences of images convey meaning. It contains four central constituents (see Figure 1): (a) the **Graphic** (phonological) **Structure**, (b) the **Narrative** (syntactic) **Structure**, (c) the **Event** (semantic) **Structure**, and (d) the **Spatial/Referential** (semantic) **Structure**.

The **Graphic Structure** is a surface-code realization of a sequence of images, *i.e.* a visual "sentence" composed of lines and shapes that serve as cues to semantic memory for their corresponding

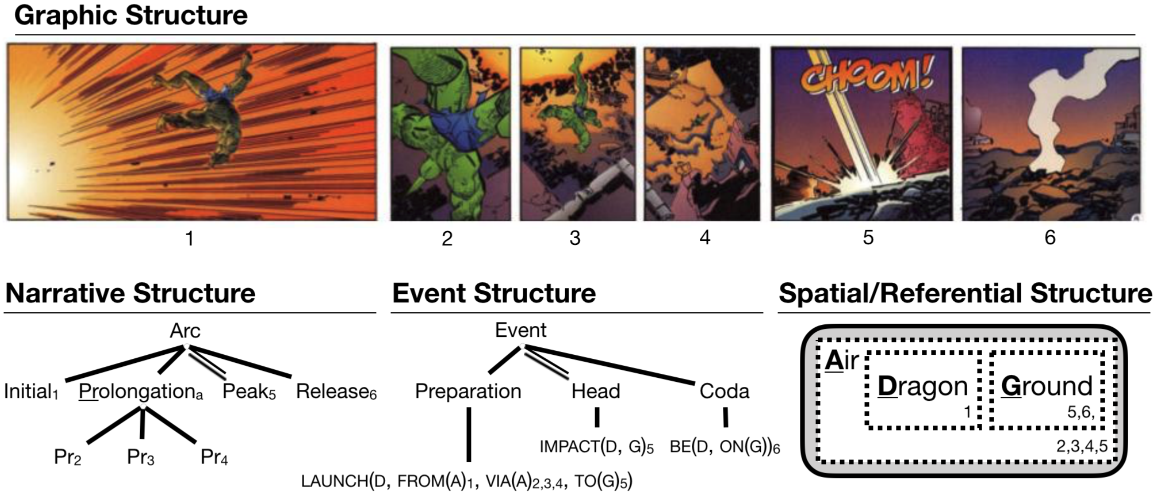

Figure 1: An illustration of elements that make up the Visual Narrative Grammar (VNG, Cohn, 2019a). The processing of phonology (Graphic Structure) happens in parallel (Jackendoff, 2007a) at both syntactic (Narrative Structure) and semantic (Event Structure and Spatial/Referential Structure) levels. All subscripts in the above diagram correspond to the Graphic Structure panels they semantically link. The example comic is a visual "sentence," with a syntactic form indicated in the Narrative Structure. The syntax structure has implications for the semantic form indicated in the Event Structure, which contains the event models of the underlying discourse. The Spatial/Referential Structure allows the semantic unification of the elements across phonological, syntactic, and semantic levels; it depends on semantic memory, which is not shown.

meaning. This visual sentence has a grammatical **Narrative Structure**, which governs the order of meaning in the depicted sequence. The example conforms to the canonical arc of visual sentences defined by Cohn (2016), a *construction grammar* (Goldberg, 2009) of the form `(Establisher) - (Initial (Prolongation)) - Peak - (Release)` with categories in parentheses being optional. Each of these categories have individual semantic implications, as do their recursive combination: in the example, the `Prolongation`$_{\mathtt{a}}$ element is a syntactic `a`(ction)-conjunction (Cohn, 2015) depicting the trajectory of a LAUNCH event. Finally, story events are grouped in a particular **Event Structure**, which partially governs the meaning of the sequence; it contains the event models (Radvansky & Zacks, 2017) built from visual narrative processing. Like the syntax of visual narrative, the semantics conform to a canonical form defined by Jackendoff (2007b), a construction grammar of the form `Preparation - Head - Coda`. In our example, the LAUNCH event constitutes the Preparation element, which relates elements that are unified within the Spatial/Referential Structure: the Dragon who was LAUNCHed FROM/VIA the Air TO the Ground. This structure integrates all the referents in semantic memory with the events within the Event Structure.

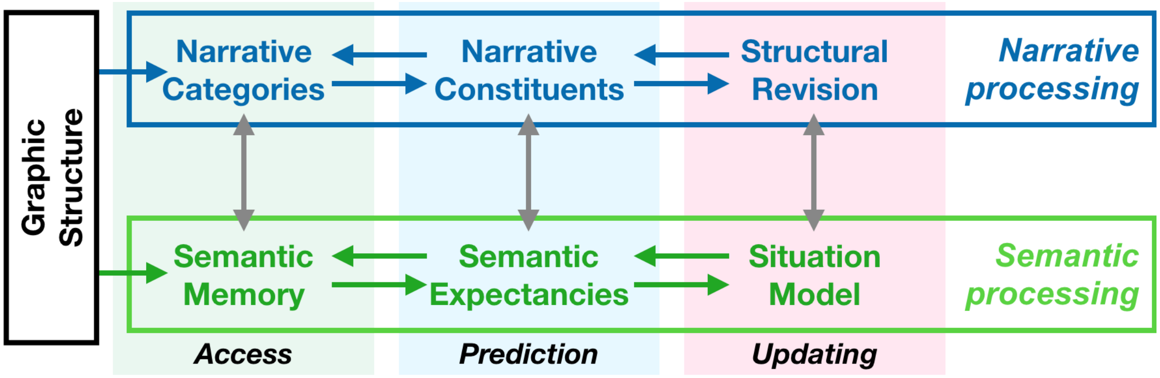

Figure 2: An illustration of the elements and interactions that make up the Parallel Interfacing Narrative-Semantics model (PINS, Cohn, 2019b). Labeled regions represent knowledge structures across narrative/semantic levels, with Access, Prediction, and Updating processes. Single-headed arrows represent feedforward/feed-backward of information, and double-headed arrows represent information interfaces between processing levels. Processing begins in parallel on the basis of some (phonological) Graphic Structure.

3.1.2 Reasoning: The Parallel Interfacing Narrative-Semantics Model

The VNPA's reasoning model, the **Parallel Interfacing Narrative-Semantics** (PINS) Model, is a theory that explains the cognitive mechanisms through which people make sense of narrative. It is illustrated in Figure 2 as a collection of integration (feedforward) and revision (feed-backward) cognitive processes involving (memory) **Access**, (structure) **Prediction** (Graesser et al., 1994), and (mental model) **Updating** (Radvansky & Zacks, 2017) that operate within and across Narrative (syntactic) and Semantic levels.

Graphic Structure images are cues to Semantic Memory and later integrated into a Situation Model based on Semantic Expectancy predictions established by an assumed *sequence continuity*: by default, we expect that an ongoing sequence will continue (semantic expectancy) and new events are created in the Situation Model when an *event discontinuity* is detected. In parallel, the semantic cues are mapped onto in-memory Narrative Categories within the canonical `(E)-(I(Pr))-P-(R)` sequence. Upon categorization, further Narrative categories (Constituents) are predicted. Revision processes are triggered when either semantic continuity prediction fails (which cues event segmentation) or narrative category prediction fails (which cues narrative structure revision).

3.2 Overview of the VNE

The Visual Narrative Engine imputes a computational representation to each of the VNG's constituent elements and a computational mechanism to a subset of the PINS methods. The mapping for VNG elements is summarized in Table 1. The PINS processes that we posit mechanisms for are the Semantic Memory Access and the Situation Model Updating. The Situation Model contains (a) the Event Model of the VNG, which we represent explicitly, (b) the Spatial/Referential information, which we represent explicitly (albeit using a different data structure), and (c) inference information

Table 1: Mapping between Conceptual Knowledge Structures in the VNG and Computational Structures in the VNE. These structures exist across the PINS processing, except for the Graphic Structure, which is the input to PINS itself.

PINS Level	**VNG Structure**	**VNE Data Structure**
N/A	Graphic Structure	Images
Narrative	Narrative Structure	Syntax Trees (via Hierarchical Decompositions)
Semantic	Event Structure	Hierarchical Plans
Semantic	Spatial/Referential Structure	Scene Graphs

between these, which we do not represent. As a starting point, we assume that the mechanism through which the Situation Model is updated already reflects modifications by its interfacing processes, which have propagated the knowledge accessed in Semantic Memory as a result of cues derived from the input Graphic Structure.

In the following sections, we elaborate on this mapping row-by-row with an explanation of the relationship and a discussion of design answers to questions we encountered when making the respective concept computationally-precise. We go over each row in the order that we feel makes sense for discussion, noting that VNPA does *not* process information in a pipeline but rather in a parallel and mutually constraining process.

VNE is implemented in the TypeScript programming language, which allows us to mathematically state the representational choices we made in our correspondence throughout this paper. When appropriate, we detail the software implementation through code snippets[1] and apply the mapping to our running example comic in Figure 1.

3.2.1 Graphic Structure as Images

While perhaps obvious, we assume that the Graphic Structure is structured as a sequence of **image** representations, composed of (for example) a two-dimensional array of pixels, with corresponding color-component intensities. VNE assumes that there is a computational model (composed of modules like those reviewed by Laubrock & Dunst) of perceptual/attentional processing that integrates with Semantic Memory and gives rise to the lexical information (images) with associated semantics.

When perceived by a reader, the Graphic Structure is used to construct the Event Structure, which represents the semantics of the visual narrative. An open question then is: how should we *begin* constructing the Event Structure? Minimally, we need know the entities involved in the event; what narratologists distinguish as the characters involved and locations in which they are set, as part of the **fabula** (Chatman, 1980). Thus, the perceptual/attentional processing model we assume above, which integrates with Semantic Memory, *must* construct an initial Spatial/Referential Structure that gets successively refined as the fabula is revealed via the (discursive) Narrative Structure. The VNE currently assumes a hand-coded Spatial/Referential Structure, discussed next.

[1] The code is available for anonymous access at `https://tinyurl.com/visual-narrative-engine`

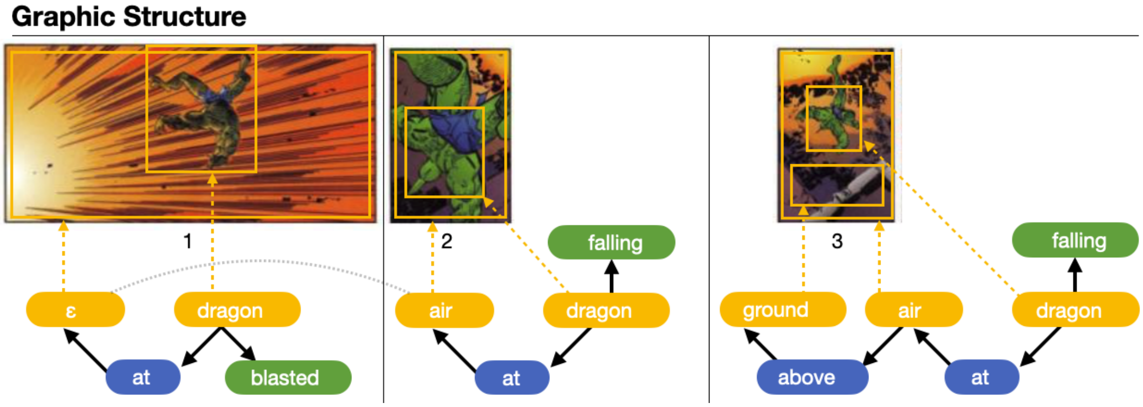

Figure 3: An example of three scene graphs and their corresponding grounding parsed from an image and the extracted logical literals. Each scene graph contains objects (*e.g.* `dragon`, in yellow), relations (*e.g.* `at`, in blue) and attributes (*e.g.* `falling`, in green).

```typescript
// Binary relations of a scenegraph
export interface RelationEdge {
    source: string, sink: string,
    relation: Relation,
    type: "RelationEdge"
}
// Unary relations of a scenegraph
export interface AttributeEdge {
    source: string,
    attribute: Attribute,
    type: "AttributeEdge"
}
type SceneGraphEdge = RelationEdge | AttributeEdge
type SceneGraph = SceneGraphEdge[]
```

Listing 1: TypeScript for the Scene Graph datatype.

3.2.2 Spatial/Referential Structures as Scene Graphs

Once constructed, the Spatial/Referential Structure consists of the reader's mental model of entities in the scenes depicted in each comic panel, as well as those entities' relationships to one another in space. A popular data structure for formalizing the contents of a depicted image, originating within the Computer Vision community, is called a **scene graph** (Krishna et al., 2016). Scene graphs have three node types. *Entities* are associations between labels and bounding boxes, such as "dragon" in Figure 3. *Attributes* are properties of entities, such as "falling." If an entity has an attribute, there is a directed edge from the entity node to the attribute node. *Relations* are binary associations between entities, such as "at." In Figure 3's second panel, the edge from "dragon" to the "at" relation, along with the edge from "at" to "air," represents "the dragon is at the air" (location). The VNE implements scene graphs as a TypeScript interface (Figure 1). Spatial semantics of a comic are expressed as a series of scene graphs, one per panel, where entities and their attributes and relations are named consistently across panels.

Two key features that VNE must support (which we do not outline here) are: (a) the creation of an initial scene graph for each comic panel, and (b) the unification of information across scene graphs. For instance, Figure 3's Panel 1 initial scene graph contains a variable ϵ (for "environment") to denote where the `dragon` is `at`, which is not immediately clear. Panel 2 illustrates the `dragon` falling downward, from which one can intuit (via an appeal to semantic memory) that the dragon is `at` the `air` (location).

Application to Running Example For the Savage Dragon comic, the attributes we define are `blasted`, `falling`, and `landed` (not shown in Figure 3). We define the relations `at` to indicate spatial containment (X is `at` Y means X is an entity spatially contained in entity Y), and `above` to represent a relative spatial orientation where the first argument is positioned above the second. Entities we reference are `dragon`, `air`, and `ground`. Using this representation, we can label each panel with a scene graph that models it (Figure 3). Given these scene graphs, we posit that there is some degree of semantic inferencing happening: the scene graph (*i.e.* state) suggests a corresponding action (*i.e.* event) which gave rise to it. These suggested events then become the basis of the Event Structure.

In codifying the scene graphs via VNE for our running example, we observed that there is a key question that remains unanswered for the study of visual narrative, which must be answered for computationally-precise simulation: "What is the relationship between the scene graph elements readers use to mentally construct the Spatial/Referential Structure and the scene graph elements authors use to render the Graphic Structure?"

Assuming (as discussed in §3.2.1) that an initial Spatial/Referential Structure is created which identifies the constituent narrative *entities* (*i.e.* characters and locations), we face a problem: nothing guarantees that the entities that are encoded by the author via the comic's phonological structure (*i.e.* the lines, shapes, and colors used) are decoded by a comic reader as intended.[2] By the same token, there is nothing that guarantees that depicted relations are interpreted correctly either. This is related to the symbol-grounding problem (Harnad, 1990), but instead of being solely concerned with how we might acquire meaning via (in our case, visual) stimuli, it concerns how comic authors might *design* (Simon, 1996) such stimuli to elicit *particular* meanings for comic readers.

In essence, we need to discover the structural relationship between (a) the ontology used to guide the authoring of the discourse and (b) the ontology used to guide the sensemaking of the discourse. Discovering this relationship would help ensure that the author's intended meaning is preserved; *i.e.* that the Spatial/Referential Structure and the Event Structure is mentally constructed in a manner the author intends. Differences in these ontologies have been studied experimentally: a person's visual language *fluency* mediates their processing of visual narrative (Cohn, 2013), and fluency evolves over time and can depend on comic reading and writing habits (Cohn, 2014). We might model this phenomenon via the precise codification of the ontologies: in our case, both the ontology of the author and audience is the logical domain of objects, relations, and attributes that are available to construct scene graphs.

The scene graphs of the Spatial/Referential Structure are (as discussed) not created in isolation. They are created as a necessary part of the Event Structure that makes up the Situation Model (*i.e.* the semantics of the comic); this Event Structure is organized per a Narrative Structure that indicates the

[2] As an anecdote: embarrassingly, several authors of this paper originally referred to the `dragon` as `Hulk`, due to the depicted character's green skin, muscular body, and cutoff shorts.

grammatical Narrative Category—*e.g.* `Establisher`, `Peak`, `Release`—of a particular scene graph (*i.e.* the syntax of the comic). In the current formulation of the VNE, the Narrative Structure is codified as part of the Event Structure. Thus, we next discuss how to precisely model the Event Structure, and how that model might be informed from the Spatial/Referential Structure codified via scene graphs.

3.2.3 Event Structures as Hierarchical Plans

Scene graphs are created via an interplay with the Event Structure, mediated by the Narrative Structure. How might we model this interplay mechanically? Before we propose an answer to this question, we note that there is an ontological difference in how the fields of cognitive psychology and neuroscience (CPN) and artificial intelligence (AI) model events.

In CPN, events are considered declarative *states of affairs* that (as detailed earlier) emerge from semantic continuity prediction failure. Events are amalgamations of spatiotemporal, causal, and entity-based information (Zwaan & Radvansky, 1998). Further, these events play a central role in human action planning (Richmond & Zacks, 2017); an action is considered the *cause* of an event continuity prediction failure, and effectively *segment* events (Newtson, 1973).

In contrast, within AI an event is a distinguished kind of action. Here, we are referring to the paradigm of **automated planning** (Ghallab et al., 2004), which models planning in a task environment as a means-ends search for a sequence of actions that transform a given state of the world into an intended goal state. The problem-solving agent (*i.e.* the planner) is described in terms of the set of actions the agent can use to transform states. Each state is a "snapshot" of a task environment: a logic-based description of the environment's configuration at some moment in time.

A planner must sometimes search for a sequence of actions in a task environment that has an internal dynamic beyond the planner's control; the planner must anticipate those dynamics in the search for a plan. In these environments, the task environment is *also* described in terms of a set of actions: namely, the state-transformations that are triggered as a consequence of the environment's dynamics. To distinguish task environment actions from agent actions, the former are referred to as *events*. Structurally, however, they are equivalent: they are both considered to represent *changes in states of affairs*. In classical automated planning specifically (*e.g.* Fikes & Nilsson, 1971), an action is represented as a pair of sets of logical conditions. The first set, termed the **preconditions**, are the conditions that must be true in the task environment's state in order for the action to execute. The second set, termed the **effects**, indicate the state-transformation that takes place as a consequence of executing the action.

A synthesis between the CPN and AI perspectives is to view an AI action as a discontinuity between two CPN events. Thus, an AI action bridges two CPN events: the precondition-based "preceding event" and the effect-based "succeeding event." We use this synthesis to posit a model of the interplay between scene graphs, the Event Structure, and its mediating Narrative Structure. Hereafter, we use the AI sense of events and actions: they are logical states of affairs and ways to effect change in these states, respectively.

From Scene Graphs to Events To go from scene graphs to events, we adopt the paradigm of *narratives as plans*. This paradigm frames story telling and sensemaking as a distinguished kind of planning process. It is used for generation due to planning affordances for representing and reasoning about causal, temporal, and hierarchical structure; it has been widely used to model the generation of plot or *fabula* (in VNPA, semantics), discourse or *sjuzhet* (in VNPA, narrative), and medium-specific narration (in VNPA, phonology) (Young et al., 2013). It is used for sensemaking due to its correspondence to cognitive theory; planning is fundamentally a search procedure and there is evidence to support the idea that narrative sensemaking involves a *search for meaning* (Graesser et al., 1994; Cardona-Rivera et al., 2016).

As observed by Cardona-Rivera & Li (2016), scene graphs are well-suited to representation as collections of logical predicates, where entities are terms, attributes are unary predicates, and relations are binary predicates. This insight, combined with our proposal to frame a planning action as a discontinuity between two events yields a way to arrive at an Event Structure from scene graphs: find the action that can best "bridge" the discontinuity across scene graphs from adjacent panels.

Narrative Planning Model: HTNs We use *hierarchical planning* (Bercher et al., 2019) as our model of narrative; specifically, we represent the Event Structure as a hierarchical task network (HTN) that is generateable by the Simple Hierarchical Ordered Planner (SHOP Nau et al., 1999). In HTN planning, actions are referred to as *tasks*, which can be primitive or compound. Primitive tasks are treated as atomic units of action, whereas compound tasks are not; these are associated with *methods*, which decompose a compound task into the subtasks needed to accomplish it. An HTN plan is a tree with method names at each internal node and primitive tasks at the leaves.

In HTN planning, the input takes the form of an initial world configuration and the name of a method to expand. In our case, world configurations are scene graphs representing the spatial-referential structure of comic panels, and the method to expand is a candidate top-level node in the event structure of the comic. Our procedure is based on PyHOP, an open-source implementation of the SHOP planner[3]. This algorithm expands each HTN method recursively, using decompositions that can apply to each successive world configuration. The search procedure successively decomposes internal nodes until primitive tasks are reached. In a departure from PyHOP, the primitive tasks are checked against the state of the world given by the scene graph; VNE continues the search until it finds a primitive task supported by the world state given in the panel's scene graph. Further, (also unlike PyHOP) instead of generating a plan as a *sequence* of actions, VNE retains the entire hierarchical structure.

Thus, VNE, unlike off-the-shelf HTN planners, simulates the human process of narrative sensemaking by taking a comic (sequence of scene graphs) as input and producing the entire HTN tree as output. The comic's depicted event structure may be read off a linearization of the leaves of the tree, but contains the entire hierarchical event structure that reflects the narrative's syntactic structure (discussed in more detail in the next section).

By analogy, consider computational parsing algorithms, which process linear input (often text) and extract a tree-structured syntax that *matches* the text. A grammar can be used to describe when a syntax tree matches a given text, but different algorithms are needed for expanding the grammar itself. As an initial proof-of-concept, our *plan matching* performs an exhaustive search through potential refinements of the HTN plan, which are presently supplied as part of the domain definition.

[3] `https://bitbucket.org/dananau/pyhop/`

Semantic Memory as an Authored HTN Domain As discussed, VNE commits to the use of a planning-like search procedure as a model of the mental process that governs the creation of the Event Structure. As part of this commitment, we need to articulate role and relevance of the planning system's ontology, or *planning domain*. In VNE, the planning domain represents the story sensemaker's Semantic Memory. The planning domain codifies (a) the predicates available to make sense of the Graphic Structure, and (b) the basic and compound tasks available to make sense of the Event Structure. Here, we only represent the ontology that guides the sensemaking of the comic, which (as alluded to) need not match the ontology that guided the comic's *generation*. Using a planning system affords the potential of using the same architecture to simulate both story-*telling* and story-*sensemaking*, to systematically explore the cognitive dynamics of reading comics.

Application to Running Example The dragon domain's primitive and composite tasks are in Listing 2 and 3, respectively. These are codified using the Hierarchical Domain Description Language (HDDL, Höller et al., 2019); the domain predicates are omitted due to space constraints.

```
;; An ?agent is launched via an explosive ?from ?thru ?to
(:action launch  :parameters (?agent - entity ?thru - path ?from ?to - loc)
  :precondition (and (at explosive ?from) (at ?agent ?from))
  :effect (and (not (at ?agent ?from)) (at ?agent ?thru) (launched ?agent ?thru ?to)))

;; A launched ?agent begins falling via ?through toward ?to.
(:action fall   :parameters (?agent - entity ?thru - path ?to - loc)
  :precondition (launched ?agent ?thru ?to)
  :effect       (and (not (launched ?agent ?thru ?to)) (falling ?agent ?thru ?to)))

;; A falling ?agent continues falling via ?through toward ?to.
(:action fall-continued   :parameters (?agent - entity ?thru - path ?to - loc)
  :precondition (falling ?agent ?thru ?to)
  :effect       (and (not (falling ?agent ?thru ?to)) (falling ?agent ?thru ?to)))

;; A falling ?a(gent) impacts ?to, and is no longer falling.
(:action impact   :parameters (?a - entity ?thru - path ?to - loc)
  :precondition (and (falling ?a ?thru ?to) (at ?a ?thru))
  :effect (and (not (falling ?a ?thru ?to)) (not (at ?a ?thru)) (landed ?a) (at ?a ?to)))

;; An ?agent continues to be at a ?place.
(:action be  :parameters (?agent - entity ?place - loc)
  :precondition (at ?agent ?place)
  :effect   (at ?agent ?place))
```

Listing 2: Savage Dragon domain: primitive task operators, written in HDDL.

In our example, VNE adds events to the Event Structure through HTN planning. The initial top-level composite task is `parse-comic-sentence` from Listing 3. The initial state may be considered to take place "offscreen," since the first panel jumps into an unstable situation, which our Semantic Memory might explain as initiated by an explosion. We codify the initial state as: `at(explosive, air)` ∧ `at(dragon, air)` ∧ `above(air, ground)`.

Given this initial state and task, the HTN planning proceeds to successively refine the plan until the tasks at the leaves of the tree are primitive and have a precondition and effect structure that matches the scene graphs from the Graphic Structure. Figure 4 illustrates the matching: the `launch`

```
;; The abstract task of parsing a comic about an ?agent.
(:task parse-comic-sentence :parameters (?agent - entity))

;; One way to parse a comic sentence about an ?agent is as: extended-fall, impact, be.
(:method sentence-parse
  :parameters (?agent - entity ?thru - path ?from ?to - loc)
  :task (parse-comic-sentence ?agent)
  :ordered-subtasks (and
    (extended-fall ?agent ?thru ?from ?to) ;; Preparation
    (impact ?agent ?to) ;; Head - Peak (Panel 5)
    (be ?agent ?to))) ;; Coda - Release (Panel 6)

;; Parse the extended-fall as a Preparation Event via the grammar rule
;; [I(initial)]+[Pr(olongation) via (a path) action-conjunction].
(:method Preparation-I+Pr-via-action-conjunction
  :parameters (?agent - entity ?thru - path ?from ?to - loc)
  :task (extended-fall ?agent ?thru ?from ?to)
  :ordered-subtasks (and
    (launch ?agent ?thru ?from ?to) ;; Initial (Panel 1)
    ;;
    ;; Prolongation-a: (action-conjunction)
    (fall ?agent ?thru ?to)             ;; Pr (Panel 2)
    (fall-continued ?agent ?thru ?to)   ;; Pr (Panel 3)
    (fall-continued ?agent ?thru ?to))) ;; Pr (Panel 4)
```

Listing 3: Savage Dragon domain: Abstract tasks and method decompositions, written in HDDL.

operator's preconditions partially match the conditions set by the scene graph in Panel 1 (the other precondition is established by the initial state we codified).

Upon termination, VNE gives the HTN plan that represents the expected mental construction of the semantics of the comic; Figure 5 illustrates this plan. However, planning is fundamentally a search algorithm, which begs the question: "how does the planner search in a way that matches human reasoning?" This is another key question that is left unanswered, and is particularly relevant to the structuring of phonology. One might imagine an Event Structure with a *different* hierarchical structure than that of Figure 5, but presently we do not know enough about the sensemaking process to understand what might yield a different parse of the Event Structure. With a planning algorithm model, we might approach a systematic way to explore this space.

3.2.4 Narrative Structure as Syntax Trees

The comic's Narrative Structure is a principal determinant of the mentally constructed Event Structure. Comics exhibit a grammatical structure that mediates the meaning of the comic. How might this be modeled mechanically, especially in light of the representation of the Event Structure as a hierarchical task network?

We propose that the Narrative Structure is codified via the HTN planning domain's *decomposition methods*, such as those presented in Listing 3. HTN decompositions can express Syntax Trees (Li et al., 2014), which give rise to the hierarchical structure that is produced via the VNE's internal HTN planning procedure. Thus, the HTN's decomposition reflects the Narrative Categories that are brought to bear during sensemaking.

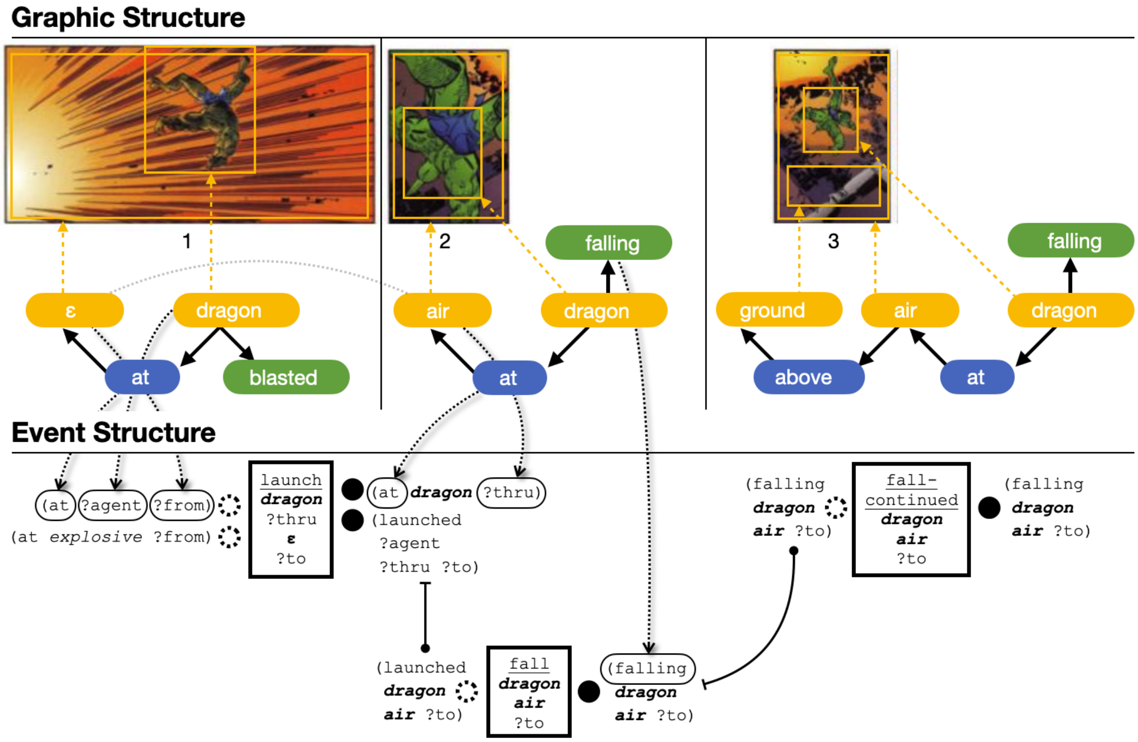

Figure 4: Events (square boxes) are added to the Event Structure via a modified HTN planning procedure: HTN plans are refined until the leaves match the scene graphs of the Graphic Structure (drawn as links to the leaf event literals). VNE also draws precondition-satisfaction links between events, drawn as circle-headed segments from the effects of one action to the preconditions of another.

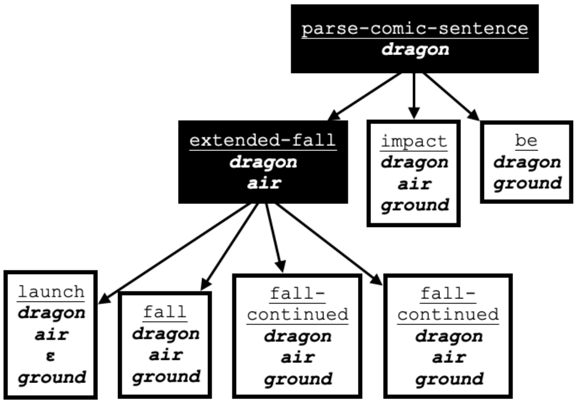
parse-comic-sentence dragon
extended-fall dragon air
impact dragon air ground
be dragon ground
launch dragon air ε ground
fall dragon air ground
fall-continued dragon air ground
fall-continued dragon air ground

Figure 5: A hierarchical plan that parses the comic in Figure 1.

Application to Running Example In our example, we only have two decompositions specified as part of the planning domain: the top-level decomposition which initiates the planning process, and an intermediate-level decomposition `extended-fall`, specified in Listing 3. The listing for `extended-fall` identifies the corresponding grammatical structure it is meant to codify: it includes the `I`nitial Panel (1) and the `P`rolongation sequence, itself a (syntactic) conjunction of panels by way of an action (the Dragon's prolonged fall through the air).

In our example, there is only *one* decomposition method so the VNE's search is trivial. However, if there is more than one option available we are left with another open question: "What guides the mental selection of particular Narrative Categories during sensemaking?" This is similar to the posed question about how to guide the HTN planning process in a manner that matches how humans do it, however the emphasis is on understanding *what* of the phonology elicits a *particular* construction of the Event Structure as mediated through the Narrative Structure. Put differently: how might we structure the phonology to syntactically elicit a *particular* Event Structure?

4. Conclusion

In summary, we developed a computational model called the Visual Narrative Engine, the first computational model of the Visual Narrative Parallel Architecture, including Visual Narrative Grammar and the corresponding Semantic Prediction operations. This work contributes an existence proof that a subset of the posited VNPA mental operations can be computationally mechanized, and it also serves to clarify where the VNPA has gaps in terms of a mechanizable theory of visual language.

While we have been able to mechanically describe the processes hypothesized to underlie the comprehension of visual narrative, we encountered (and throughout the paper, identified) several open questions that *must* be answered for the VNE to fully simulate all cognitive processes posited in the VNPA. These questions center on (a) the degree of correspondence between the logical language used to generate comics and the language used to make sense of them and (b) the fact that, because Event Structure-building is structured as a planning process, and planning is a search algorithm, we need to be more precise about the control knowledge that governs how the search proceeds. For the first question, we feel that exploring what knowledge structures are created in the mind as a consequence of visual language fluency will be a productive research program. For the second question, we feel that exploring how the same sequence of comic panels might be interpreted differently will shed insight into what kinds of algorithms humans use to search for the semantic meaning of depicted phonology. We expect the VNE to play a significant epistemological role in the creation of theories that aim to characterize visual narrative processing writ large.

In future work, we plan to investigate the VNE's potential as the foundation for tools to support creative human-computer interaction. The history of programming languages takes a crucial turn after the understanding that human intuition for language compositionality can be formalized as context-free grammar, enabling the development of higher-level languages understandable by both humans and machines. Analogously, we plan to explore the extent to which *shared visual language* can be developed for human-computer collaboration. This work will require being able to anticipate the cognitive effects of visual narratives on their audiences, including *inferences* Iyyer et al. (2017) prompted by the strategic removal or obscuring of visual information Cohn (2019a). Potential applications include "interactive comics," storytelling experiences for which visual scenes may be computationally generated, as well as mixed-initiative authoring tools for human-computer co-creative comic creation.

Acknowledgements

We gratefully acknowledge the reviewers for their thorough and constructive comments, as well as the University of Utah School of Computing for providing collaboration space for authors Martens and Cardona-Rivera.

References

Bercher, P., Alford, R., & Höller, D. (2019). A survey on hierarchical planning–one abstract idea, many concrete realizations. *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (pp. 6267–6275).

Cardona-Rivera, R. E., & Li, B. (2016). Plotshot: Generating discourse-constrained stories around photos. *Proceedings of the 12th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (pp. 2–8).

Cardona-Rivera, R. E., Price, T. W., Winer, D. R., & Young, R. M. (2016). Question Answering in the Context of Stories Generated by Computers. *Advances in Cognitive Systems*, *4*, 227–246.

Chatman, S. B. (1980). *Story and Discourse: Narrative Structure in Fiction and Film*. Cornell University Press.

Cohn, N. (2013). *The visual language of comics: Introduction to the structure and cognition of sequential images.*. A&C Black.

Cohn, N. (2014). *The visual language fluency index: A measure of "comic reading expertise"*. Visual Language Lab. From `www.visuallanguagelab.com/resources.html`.

Cohn, N. (2015). Narrative conjunction's junction function: The interface of narrative grammar and semantics in sequential images. *Journal of Pragmatics*, *88*, 105–132.

Cohn, N. (2016). *The visual narrative reader*. Bloomsbury Publishing.

Cohn, N. (2019a). Being explicit about the implicit: inference generating techniques in visual narrative. *Language and Cognition*, *11*, 66–97.

Cohn, N. (2019b). Your brain on comics: A cognitive model of visual narrative comprehension. *Topics in Cognitive Science*.

Cohn, N., & Magliano, J. P. (2019). Editors' introduction and review: Visual narrative research: An emerging field in cognitive science. *Topics in Cognitive Science*.

Fikes, R. E., & Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, *2*, 189–208.

Gerrig, R. J., & Bernardo, A. B. I. (1994). Readers as problem-solvers in the experience of suspense. *Poetics*, *22*, 459–472.

Gervás, P. (2009). Computational approaches to storytelling and creativity. *AI Magazine*, *30*, 49–62.

Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated Planning: Theory & Practice*. Elsevier.

Goldberg, A. E. (2009). The nature of generalization in language. *Cognitive Linguistics*, *20*, 93–127.

Graesser, A. C., Singer, M., & Trabasso, T. (1994). Constructing inferences during narrative text comprehension. *Psychological review*, *101*, 371–395.

Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, *42*, 335–346.

Höller, D., Behnke, G., Bercher, P., Biundo, S., Fiorino, H., Pellier, D., & Alford, R. (2019). Hddl–a language to describe hierarchical planning problems. *arXiv preprint arXiv:1911.05499*.

Iyyer, M., Manjunatha, V., Guha, A., Vyas, Y., Boyd-Graber, J., Daume, H., & Davis, L. S. (2017). The amazing mysteries of the gutter: Drawing inferences between panels in comic book narratives. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7186–7195).

Jackendoff, R. (2007a). A parallel architecture perspective on language processing. *Brain research*, *1146*, 2–22.

Jackendoff, R. (2007b). Shaking hands and making coffee: The structure of complex actions. In *Language, consciousness, culture: Essays on mental structure*, chapter 4. MIT Press.

Krishna, R., et al. (2016). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv*. From `https://arxiv.org/abs/1602.07332`.

Langley, P. (2012). The cognitive systems paradigm. *Advances in Cognitive Systems*, *1*, 3–13.

Larsen, E. (1993). *Savage dragon*, volume 1. Image Comics.

Laubrock, J., & Dunst, A. (2019). Computational approaches to comics analysis. *Topics in Cognitive Science*, *n/a*.

Li, N., Cushing, W., Kambhampati, S., & Yoon, S. (2014). Learning probabilistic hierarchical task networks as probabilistic context-free grammars to capture user preferences. *ACM Transactions on Intelligent Systems and Technology*, *5*, 1–32.

Lieto, A., & Radicioni, D. P. (2016). From human to artificial cognition and back: New perspectives on cognitively inspired ai systems. *Cognitive Systems Research*, *39*, 1 – 3.

Marsella, S. C., & Gratch, J. (2009). Ema: A process model of appraisal dynamics. *Cognitive Systems Research*, *10*, 70–90.

Martens, C., & Cardona-Rivera, R. E. (2016). Generating abstract comics. *Proceedings of the 10th International Conference on Interactive Digital Storytelling* (pp. 168–175). Springer.

McNamara, D. S., & Magliano, J. (2009). Toward a comprehensive model of comprehension. *Psychology of learning and motivation*, *51*, 297–384.

Mueller, E. T. (2013). Computational models of narrative. *Sprache und Datenverarbeitung: International Journal of Language Processing*, *37*, 11–39.

Nau, D., Cao, Y., Lotem, A., & Munoz-Avila, H. (1999). SHOP: Simple Hierarchical Ordered Planner. *Proceedings of the 16th International Joint Conference on Artificial intelligence* (pp. 968–973).

Newtson, D. (1973). Attribution and the unit of perception of ongoing behavior. *Journal of Personality and Social Psychology*, *28*, 28.

Radvansky, G. A., & Zacks, J. M. (2017). Event boundaries in memory and cognition. *Current opinion in behavioral sciences*, *17*, 133–140.

Richmond, L. L., & Zacks, J. M. (2017). Constructing experience: Event models from perception to action. *Trends in cognitive sciences*, *21*, 962–980.

Saraceni, M. (2016). Relatedness: Aspects of textual connectivity in comics. In N. Cohn (Ed.), *The visual narrative reader*, chapter 5, 115–128. Bloomsbury.

Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and brain sciences*, *3*, 417–424.

Simon, H. A. (1996). *The sciences of the artificial*. MIT press, 3rd edition.

Sun, R. (2008). Introduction to computational cognitive modeling. In *Cambridge handbook of computational psychology*, chapter 1, 3–19. New York, NY, USA: Cambridge University Press.

Young, R. M., Ware, S., Cassell, B., & Robertson, J. (2013). Plans and Planning in Narrative Generation: A Review of Plan-Based Approaches to the Generation of Story, Discourse, and Interactivity in Narratives. *Sprache und Datenverarbeitung*, *37*, 67–77.

Zwaan, R. A., & Radvansky, G. A. (1998). Situation models in language comprehension and memory. *Psychological bulletin*, *123*, 162.