

# Programming Models for Interactive Fiction

Ethan Robison

## Natural Language, Semantic Analysis and Interactive Fiction

```
@article{nelson2006natural,  
title={Natural language, semantic analysis, and interactive  
fiction},  
author={Nelson, Graham},  
journal={IF Theory Reader},  
volume={141},  
pages={99-104},  
year={2006}}
```

**Summary:** Nelson makes an argument for using natural language<sup>1</sup> as the language of choice for both authoring and displaying interactive fiction experiences. He outlines the impact of this choice in the implementation and common use of the interactive fiction software Inform.<sup>2</sup>

Natural language is (1) presumably already within the grasp of the programmer (in the general sense, but particularly in the sense of the interactive fiction programmer),<sup>3</sup> (2) more robust to changes in internal structures of the program, and (3) a good way of capturing the behavior of a particular program, since individuals communicating about a particular program are likely to use natural language to do so.<sup>4</sup> Furthermore, the criticisms of natural language for programming, that it is ambiguous and verbose,<sup>5</sup> are, Nelson argues, less founded than they might seem. The ambiguity is avoided by changing diction—that is, a programmer can change the *way* in which they “speak” to be more specific. The verbosity is sidestepped by using linguistic features like reference (both anaphoric and cataphoric) and, again, a fluidity of allowed diction.

---

<sup>1</sup>Specifically, a subset of English, but one which is hopefully well-chosen.

<sup>2</sup><http://inform7.com/>

<sup>3</sup>This is not a commentary on which *specific* natural language might come to this programmer, although the Inform designers chose English as their language of choice.

<sup>4</sup>Nelson isn’t arguing that formalisms are bad, and acknowledges that, under the hood, Inform7 has to run just like any other program; rather, he’s suggesting that the advantages afforded by formal languages are outweighed by those afforded by natural language.

<sup>5</sup>Yes, there are others, but these two are particularly frustrating to programmers. Or, at least, to me.

Nelson covers also the structure of Inform's ontology, both by default and how it might be extended or modified by the programmer. He also covers modifications to the behavior of the game at runtime. These are ancillary to the thesis of the work, however.

**Evaluation:** This paper is foundational to understanding *why* Nelson would choose to use natural language for programming when intuition would suggest otherwise. Though it does not cover the specifics of the implementation,<sup>6</sup> the paper talks instead about the specific kinds of problems that Inform7's infrastructure is trying to solve (reference, rules, exceptions to rules, etc.).

It is hard to say how impactful this specific paper was in the world of interactive fiction. It is unlikely that would-be content creators would choose Inform over other software because of Nelson's work in writing this paper; rather, they would choose it for the features of Inform that he then outlined in this paper. In other words, while this paper is a summary of what makes Inform great, it would be great whether he wrote it or not. In the end, this paper is probably better at convincing the technically-minded audience of the value of Inform's choice to be written like natural language than at convincing its end-users of its greatness. The proof is in the pudding for Inform, and this paper just serves to convince the doubtful to give that pudding a try.

---

<sup>6</sup>In fact, finding formal definitions of Inform7's structure is surprisingly difficult, from a programmer's perspective. Then again, it's not really for us, is it?