SHRDLU: A Game Prototype Inspired by Winograd's Natural Language Understanding Work

Santiago Ontañón
Drexel University, USA
so367@drexel.edu

Abstract

This paper describes a game prototype called "SHRDLU" that explores the concept of designing a game around the ideas behind Winograd's original SHRDLU system. We briefly describe the main gameplay, as well as the natural language and inference architecture used by game NPCs.

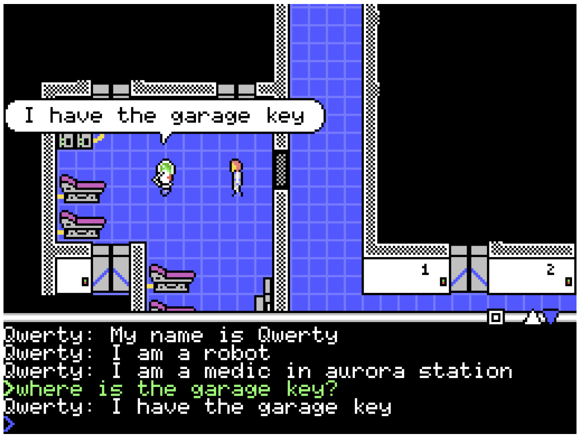
Introduction

This paper describes a game prototype designed around ideas behind Winograd's original SHRDLU system. Specifically, we present an early prototype of "SHRDLU", an adventure game where the player controls a character in a sci-fi setting world, and can talk to the game NPCs in plain natural language. The name of the game comes from the fact that NPCs behave in a similar manner to Winograd's SHRDLU (Winograd 1972) (in the rest of this paper, we will use SHRDLU to refer to our game and "Winograd's SHRDLU" to refer to the original AI system). These NPCs interact in natural language, and have reasoning capabilities beyond what is usual in standard NPCs in this type of games, thanks to a full first-order logic resolution engine.

SHRDLU

SHRDLU[1] is a sci-fi adventure game with a gameplay that borrows ideas from the early *Sierra* graphic adventure games of the late 80s such as "Space Quest" (Crowe and Murphy 1986). The player finds itself in a space station in an unknown planet without remembering how did she get there. By interacting with three robot characters (Etaoin, Qwerty and Shrdlu), the player will unravel the mystery about her identity and about Aurora Station.

As shown in Figure 1 the key difference between SHRDLU and Sierra adventure games is that, while in Sierra adventures the player had to type commands to make the main character perform actions (like in earlier Zork-style text-based adventures), in SHRDLU typing is used to talk to other NPCs in a similar way as in Façade game (Mateas and Stern 2003), or MKULTRA (Horswill 2014).



I have the garage key

```
Qwerty: My name is Qwerty
Qwerty: I am a robot
Qwerty: I am a medic in aurora station
>where is the garage key?
Qwerty: I have the garage key
>
```

Figure 1: A screenshot of SHRDLU showing the main character talking with an NPC robot named Qwerty.

Different from Façade, where the focus was on balancing user and author intent to maintain an emotionally intense, dramatic action, SHRDLU employs deep natural language parsing and generation. All of the text produced by the game NPCs is automatically generated from their internal logical representations, and thus, they do have a logical representation of what they are saying which they can reason about. A more related game is Ian Horswill's MKULTRA, also based around natural language processing inspired by Winograd's SHRDLU, but with different game mechanics (such as belief injection, to solve puzzles by making NPCs perform the actions the players wants).

The AI Behind the NPCs in SHRDLU

Winograd's SHRDLU was an early AI system that allowed the user to interact with an avatar in a blocks world using natural language. SHRDLU was able to solve complex tasks via planning (problem solving), where the user could ask SHRDLU to perform a complex task, and SHRDLU would have to plan, Additionally, SHRDLU was also able to answer questions about the world, or about the reasons for performing certain actions. In addition to the player character, SHRDLU features three robot NPCs. Each NPC is controlled by a separate Winograd's SHRDLU-style AI system

[1] SHRDLU can be found here https://github.com/santiontanon/SHRDLU

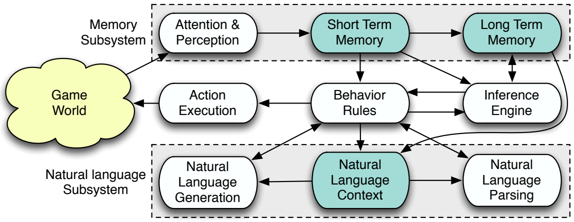


Figure 2: Main components of the AI architecture for the NPCs in SHRDLU.

with both problem solving and question answering capabilities. Figure 2 shows the main modules of this AI system. Let us briefly describe each of the modules in turn.

Memory Subsystem: SHRDLU uses a typed first-order logic knowledge representation to represent facts (e.g. "a table is a piece of furniture") and inference rules (e.g., "if an object is in a location X, and X is in Y, the object is also in Y"). This representation was designed specifically for the game and extends standard first-order logic by assuming the existence of an ontology O where each possible functor or constant is organized in a multiple-inheritance hierarchy.

Every NPC in the game has a selective attention procedure (which makes the game both more realistic, and also keeps the knowledge base manageable). All clauses representing perception are stored in a *short term memory* buffer, where they stay for only a fixed amount of time before they are forgotten. Certain perceptions (such as verbal utterances) are stored in long-term memory, which also contains background knowledge, and all the facts learned from dialog with other NPCs or with the player (the AI does not distinguish when they are talking to the player or to another NPC).

Inference: NPCs are equipped with a full *resolution* engine (Robinson 1965), which is sound and complete for the typed first-order logic used in the game. For example, if the player asks "where is Qwerty?" to some NPC, a resolution process is started trying to see if there is any value of X for which the clause $location.at(`qwerty'[\#id], X[location])$ is entailed by the knowledge base (values between square brackets represent the types of the attributes in the term).

Behavior Rules: SHRDLU uses a formalism based on *speech act* theory (Searle 1969) to represent the different utterances that players or NPCs can say in natural language. NPCs in the current version of SHRDLU can recognize and generate 30 different speech acts that include things like: *greet*, *inform*, *acknowledge*, *request action*, *where question*, *when question*, etc. A collection of "behavior rules" dictate how the different internal memory structures of an NPC are updated upon receiving each different speech act, and if any action needs to be undertaken as a response.

Natural Language Subsystem: This is the most complex component of SHRDLU's NPC AI architecture and is organized in three main modules:

Natural Language Context: this is a record of the current state of a conversation, including the set of speech acts exchanged till now, and the set of objects referred in them. Additionally, it records whether the NPC is waiting for a response, expected to thank or acknowledge anything, provide a response to any question, etc. This context is used to disambiguate pronouns or referring expressions in natural language, e.g. if, for example, if the player says "give me the key", but there are two keys currently in the room. If only one of the keys appears in recent speech acts in the conversation, then the NPC determines that is the key the player is referring to.

Natural Language Parsing: the goal of this module is to translate a natural language sentence into a speech act. The module uses a grammar which defines a collection of parsing rules for each speech act (and collections of generic rules for noun phrases, verb phrases, etc. that are used by the other rules). The key differences between this custom parser with respect to standard NLP parsers such as the Stanford Parser (Klein and Manning 2003) are: 1) the output of parsing is not a parse tree of the sentence, but a logical representation of the speech act (e.g., "the red key" will get translated to the identifier "key1", and the fact that the natural language sentence used the adjective "red" is lost); 2) we call this a "situated parser", as parsing rules include calls to special *de-reference* functions, which search for objects in the natural language context that match with referring expressions such as "the red key", or "this door". There are three types of de-reference functions (context dereference, hypothetical dereference and query dereference) that are used to translate phrases into logical descriptions.

Natural Language Generation: translates speech acts represented in logical form to natural language. Each speech act has its specialized routine for this process. Every single natural language sentence produced by any NPCs is generated using this module.

Conclusions and Future Work

This paper has presented SHRDLU, a game prototype exploring the type of gameplay that can be achieved by using Winograd's SHRDLU-style NPC AI. The current version of the prototype includes the whole first "act" of the game (out of a total of 4 for the complete storyline). Although no systematic nor thorough evaluation has been conducted at this point, initial evaluation with members close to our research team indicates that players are able to communicate with the NPC to get them to do what they need, and can get the answers necessary to progress in the story. We have also seen that the constrained story world of the game is, while much broader than the simple blocks world of Winograd's SHRDLU, still constrained enough for making natural language parsing feasible with a high degree of success.

As part of our future work, we plan to expand the set of speech acts that is currently recognized by the natural language subsystem. For example, even if the inference system allows it, there is currently no parsing rules for speech acts corresponding to asking "why" questions, which was one of the main features of Winograd's SHRDLU.

Acknowledgements: We would like to thank the people that helped beta test the current prototype: Jichen Zhu, Jordi Sureda, Josep Valls-Vargas, and Sam Snodgrass.

References

Crowe, M., and Murphy, S. 1986. Space Quest: Chapter I – The Sarien Encounter. *Sierra On-Line*.

Horswill, I. 2014. Game design for classical ai. In *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Klein, D., and Manning, C. D. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st annual meeting of the association for computational linguistics*.

Mateas, M., and Stern, A. 2003. Façade: An experiment in building a fully-realized interactive drama. In *Game developers conference*, volume 2, 4–8.

Robinson, J. A. 1965. A machine-oriented logic based on the resolution principle. *Journal of the ACM (JACM)* 12(1):23–41.

Searle, J. R. 1969. *Speech acts: An essay in the philosophy of language*, volume 626. Cambridge university press.

Winograd, T. 1972. Understanding natural language. *Cognitive psychology* 3(1):1–191.