SHRDLU

Nigel Ward, University of Tokyo, Japan

CONTENTS

SHRDLU was an integrated artificial intelligence system could make plans and carry on simple conversations about a set of blocks on a table.

INTRODUCTION

00056001 SHRDLU, created by Terry Winograd, was a computer program that could understand instructions and carry on conversations about a world consisting of some toy blocks on a table. For example:

Person: ‘*Pick up a big red block*.’
Computer: ‘OK.’ (In the display, SHRDLU’s animated robot hand does this.)
Person: ‘*Grasp the pyramid*.’
Computer: ‘I DON’T UNDERSTAND WHICH PYRAMID YOU MEAN.’
Person: ‘*Find a block which is taller than the one you are holding and put it into the box*.’
Computer: ‘BY “IT” , I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE ONE I AM HOLDING.’
Computer: ‘OK.’ (Does it.)
Person: ‘*What does the box contain?*’
Computer: ‘THE BLUE PYRAMID AND THE BLUE BOX.’
. . .

00056002 SHRDLU was perhaps the first integrated artificial intelligence (AI) system. It incorporated modules for parsing, planning, question-answering, dialogue management, etc., which worked together to perform a complete task.

PLANNING

00056003 SHRDLU controlled a simulated robot hand with which to arrange the blocks as the user requested. This was not trivial, since the robot could directly perform only three actions: `MOVETO` (a location), `GRASP` (a block) and `UNGRASP` (the block currently in the hand). Thus SHRDLU, which accepted high-level commands, such as ‘put a red block on the green block’, had to discover sequences of basic actions that would achieve its goals. It did this with a ‘Planner’ program.

The key idea, as in many planning systems, was a 00056004
technique now called backward chaining or means-ends analysis. For example, if the goal was to have ‘a red block on the green block’, the Planner would first check to see if this was true already, and if not, it would try to find a way to achieve it. To find a way, it would search through the various operations in its knowledge base. In this example, this would yield `UNGRASP` as a possible way to satisfy the goal, provided that three preconditions held: (1) the green block had nothing on top, (2) the hand was holding a red block, and (3) the hand was over the green block. The Planner would then check to see if these preconditions were true, and if not, develop sub-plans to achieve each precondition. For example, if the green block had something on top, the Planner would proceed to find an operation or plan to achieve the ‘nothing-on-green’ goal. The concatenation of all the operations found in this process gave the sequences of actions needed to achieve the final goal.

SHRDLU also kept track of these chains of 00056005
reasoning, so that it if the user later asked, for example, ‘Why did you pick up the brown pyramid?’. SHRDLU could give an answer like ‘SO THAT THE GREEN BLOCK WOULD BE CLEAR SO I COULD PUT THE RED BLOCK ON IT’.

SHRDLU was also able to backtrack when a par- 00056006
tial plan turned out to be infeasible. That is, it was able partially to undo some line of reasoning, and start looking for an alternative sub-plan without having to start again from scratch.

PROCEDURAL SEMANTICS

Planner used a special internal representation. 00056007
Thus, ‘a red block in the box’ would be represented as something like

```
(THGOAL (#IS ?X #BLOCK)) (THGOAL
(#COLOR $?X #RED)) (THGOAL (#IN $?X
:BOX))
```

00056008 With some familiarity with the conventions of logic, and a little imagination, it should be possible for an intelligent reader to see what this means. However, for this to be meaningful for a program requires something more, Winograd noted. SHRDLU needed to be able to reason about the basic concepts such as `#RED` and `#IN`. At the time, the representation of reasoning as theorem-proving, in which formal procedures manipulated logical symbols without using any knowledge of what the symbols 'meant', was prevalent among AI researchers. Winograd, however, argued for a 'procedural' view of semantics. In a vivid illustration of this approach to meaning, Winograd showed how the concept `CLEARTOP`, corresponding to the English phrase 'clear off', can be expressed as a procedure:

1. Start.
2. Does X support an object Y? If not, go to 5.
3. Move Y off X.
4. Go to 2.
5. Assert that X is `CLEARTOP`

00056009 Similarly, the meaning of 'a red cube which supports a pyramid' can be described as a procedure for looping through all pairs of blocks on the table while checking their dimensions, colours, and relations to each other. This viewpoint, called 'procedural semantics', sparked excitement at the time: it seemed that this fresh perspective, inspired by practical programming considerations, would allow AI to resolve or bypass the messy problems that had troubled philosophers of meaning for centuries. In practice, however, most of the procedural knowledge in SHRDLU was represented as simple declarative facts, goals, or operators, and a sharp distinction between procedural and declarative approaches turned out to be both elusive and unimportant.

00056010 Building SHRDLU involved a lot of clever programming, and not just the implementation of a mathematical or psychological theory. For this reason, SHRDLU is a landmark 'hacker' AI system. Even today, a lecture about SHRDLU is a common way to excite and inspire computer science undergraduates. However, the 'hacker' approach tends to lead to messy programs and confused models, and modern AI researchers are generally more careful to distinguish between the systems-building and the theory-building aspects of AI.

ANALYSING THE INPUT

Since SHRDLU's inputs were English sentences, it 00056011
first had to parse them: to work out where the noun phrases began and ended, which words went with with which, and so on.

In designing the language analysis model, Wino- 00056012
grad looked beyond purely practical considerations. Reasoning that in a psychologically realistic grammar, syntax and semantics were interdependent, he chose to base SHRDLU's grammar on systemic grammar. Winograd did not believe that it was possible or desirable to have an autonomous theory of syntax, formulated without regard to meaning.

SHRDLU also embodies a highly 'interactionist' 00056013
view of the relation between syntactic and semantic processing. For example, consider the command 'Put the blue pyramid on the block in the box', which is ambiguous (as to whether 'on the block' describes the current location of the pyramid or its desired location). For this sentence, SHRDLU would first recognize that 'the blue pyramid' was a syntactically possible noun phrase, then immediately check to see if this interpretation made sense semantically. Because SHRDLU knew about the locations of all the blocks, it could determine whether there was a unique blue pyramid. If so, the parser would know that the rest of the sentence, 'on the block in the box', had to be a single prepositional phrase, specifying where to put the pyramid. If, on the other hand, there was more than one blue pyramid, the parser would deduce that 'on the block' was part of the noun phrase; then SHRDLU would perform another semantic check, to make sure that 'the blue pyramid on the block' was a valid description of a unique object.

In this way, SHRDLU used syntactic and seman- 00056014
tic information in concert to efficiently arrive at the correct interpretation of the input: by using semantic information early, it avoided wasting time considering interpretations which were syntactically possible but actually impossible given the configuration of the blocks. Thus SHRDLU demonstrated that an AI system with multiple modules could be tightly integrated, for the sake of both cognitive plausibility and efficiency. At the same time, SHRDLU was a fairly modular system, with the different kinds of data structures and processing algorithms neatly separated into components, in accordance with general software engineering principles, making the system easier to develop, debug, and extend. Of course, there is something of a trade-off between integration and modularity,

and dealing with this was a central concern of the new sub-field of 'AI architectures' for many years.

RESEARCH STRATEGY

00056015 In 1972 SHRDLU was an amazing achievement. It succeeded partly because it worked in a 'microworld', that is a limited domain, consisting of some coloured blocks on a table. It could only understand sentences about blocks, only make plans about blocks, and only ask questions about blocks. This may seem like a retreat from real-world complexities, but Winograd argued that it was a reasonable first step. After all, small children play with blocks before going on to more complex things. Similarly, AI should perhaps start by developing systems for various microworlds. Each system would be a simple but complete application. As time went on, more complex microworlds would be tackled, and the inventory of AI knowledge and techniques would grow, and ultimately be combined. This approach is not universally accepted, and many have argued that AI for the real world cannot be developed by this strategy. Nevertheless, today all useful AI systems operate in microworlds. Indeed, the successful deployment of AI techniques is known to require the identification of some domain of expertise (such as engine diagnosis, chess playing, or answering payroll questions) that can be dealt with in isolation from the unbounded complexity of the 'real world'.

00056016 SHRDLU's microworld allowed direct mappings from lexical categories (in the input sentences) to meaning elements, and from syntactic structures to rules for combining meaning elements, as seen in table 1. (Of course language is more complex than this in the real world: for example, '*destruction*' is a noun, but refers to an activity, not an object; and '*fake gun*' does not refer to the subset of guns that are fake.)

DIALOGUE MANAGEMENT

00056017 Perhaps the most appealing aspect of SHRDLU was that it could hold a reasonably involved conversation and not lose the thread – something that is not trivial even for people. For example, in the dialogue above Shrdlu deals with words like 'it' and 'that' which refer back to things it heard or did or said earlier. SHRDLU was also able to formulate follow-up questions if the input was ambiguous. As such, it was probably the first system to incorporate what is now called a 'dialogue manager' and face up to the problems that arise when dealing with discourse, not just individual sentences.

IMPACT AND LIMITATIONS

SHRDLU was one of the first AI systems to perform 00056018
a realistic task. Although moving coloured blocks around on tables is not a commercially important activity, SHRDLU-like dialogues are very plausible when querying databases. Thus SHRDLU inspired, in the early 1970s, one of the periodic rushes to commercialize AI. There were several efforts to build 'natural language interfaces' to databases, so that people could get the information they wanted (for example, 'a list of all single women in zip code 94705 with cats' or 'a list of all people who earn more than their managers') without having to write programs or arcane database queries.

This turned out to be easy to achieve for certain 00056019
hand-chosen inputs, but not for the sorts of things that real users input. To build a natural language interface for a new domain, such as a shoe shop inventory database, would require months of programming effort, and even then users tended to lapse into a narrow subset of English after discovering that fragmentary, idiomatic or unusual phrasings were seldom understood correctly. The nail in the coffin for natural language interfaces was the invention of graphical user interfaces (GUIs). Most users prefer GUIs, with on-screen views of the data and with the available commands laid out in buttons and menus, rather than natural language. Today, however, there is a revival of interest in natural language interfaces for spoken language interaction: for example, to get airline flight information over the telephone. Systems capable of carrying on rudimentary conversations about such information are now becoming common.

SHRDLU was also one of the first AI systems to 00056020
capture the popular imagination. Winograd's style of research, taking inspiration from linguistics and psychology and using low-level programming tricks, was inaccordance with the intellectual climate of the time. The view he propounded – that programming considerations can tell us how a human reasoning system might work, and conversely, that knowledge of human cognitive processing could inform AI system development – implied that AI researchers, linguists and psychologists could achieve great things if they worked together. This excitement helped to stimulate the new field of cognitive science.

00056t001 ## Further Reading

Winograd T (1972) *Understanding Natural Language*. Location: Academic Press. Also published in *Cognitive Psychology* **3**(1): 1–191.

Winograd T (1973) *A procedural model of language understanding* In: Schank R and Colby K (eds) *Computer Models of Thought and Language* pp 152–186. New York, NY: WH Freeman [Reprinted in: Grosz BJ, Jones KS and Webber BL (eds) (1986) *Readings in Natural Language Processing*. Los Altos: Morgan Kaufmann.

Jurafsky D and Martin JH (2000) *An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Location Prentice-Hall.

Boden MA (1987) *Artificial Intelligence and Natural Man*, 2nd edn. Location MIT Press.

Marr D (1977), Artificial intelligence – a personal view. *Artificial Intelligence* **9**: 37–48.

McDermott D (1981) Artificial intelligence meets natural stupidity. In: Haugeland J (ed) *Mind Design: Philosophy, Psychology, Artificial Intelligence* pp 143–160. Location MIT Press. *SHRDLU*. http://hci.stanford.edu/cs147/examples/shrdlu/

Glossary

Concatenation to join two or more pieces by simply placing them adjacently, as in concatenating ‘real’ and ‘time’ to get ‘realtime’.

Dialogue management In a system that interacts with the user over several conversational turns, the process of keeping track of the context and of deciding what to say next

Internal representation A data structure that contains the information an intelligent system needs to act, typically a symbolic representation of some aspects of the external world

Microworld A limited or simplified domain in which a system can display some intelligence or competence without being robustly or generally intelligent.

Modular design A complex system built out of separate components, or modules, each of which is mostly self-contained and interferes with the other modules as little as possible. Such designs are often easier to design, develop, debug, and extend.

Procedural semantics The use of procedures (short abstract computer programs), instead of declarative statements, to represent meaning or knowledge.

Recursion an algorithm which solves a complex problem by breaking it down into smaller pieces and then solving them using the same algorithm, recursively.

Keywords: (Check)

natural language; question-answering; dialogue management; knowledge representation; integrated AI system

Table 1. Some semantic interpretation rules exploited in SHRDLU

Language	*Meaning*
noun	object
adjective	a property of an object
‘*the*’	expect to find a unique object
verb	action
‘*it*’	the most recent topic of conversation
question mark	compute the answer
adjective – noun	a subset: all members of the set that have the specified property
verb – noun phrase	perform the action on the set of objects specified by the noun phrase

Typesetter’s Query

tq1 Level number not available