

## Referential Nets as ACT-R Declarative Memory Representation

Markus Guhe (m.guhe@ed.ac.uk)

Human Communication Research Centre & Linguistics and English Language, University of Edinburgh  
Adam Ferguson Building, 40 George Square, Edinburgh EH8 9LL, UK

### Introduction

While computational cognitive modelling within unified theories of cognition (UTC, Newell 1990) has made impressive strides throughout the last two decades, it still hosts little research on language. (Notable but rare exceptions are Lewis 1993 and Budiu & Anderson 2004.) If language is what makes humans human, it is difficult to see how our cognitive models can come close to a comprehensive understanding of human cognition and intelligence without graceful inclusion of language.

On the other hand there is the large field of computational linguistics (CL), which successfully builds computational models of language – albeit mostly non-cognitive models. Each field exists independently of the other, largely because of different traditions and tools: CL being a child of AI and linguistics and UTC a child of AI and psychology. And while CL mainly uses logic-based approaches to explain language as a system, UTCs tend to be production systems to simulate cognitive phenomena other than language, e.g. memory, reasoning, categorisation. Although one could expect psycholinguistics to fill this gap, this discipline only rarely concerns itself with modelling, and if it does, it usually uses one-off models.

In this situation it is desirable to bridge the gap between CL and UTCs by finding a method to transfer research results from CL to UTCs. The assumption underlying this attempt is that non-cognitive models of language do shed light on the cognition of language; an assumption that is supported, for example, by Reiter's (1994) analysis, which shows a large degree of congruence between cognitive and non-cognitive models of language, and by the successes of CL models to account for cognition, e.g. Latent Semantic Analysis (Landauer, Foltz & Laham 1996).

### Referential Nets and ACT-R

To provide well-formed linguistic representations, which can serve the comprehensive ACT-R approach (Anderson & Lebiere 1998), I will present a preliminary account how *referential nets* can be translated into the chunk-based representations of ACT-R's declarative memory. (Habel 1986 defines referential nets; Guhe, Habel & Tschander 2003 and Guhe in press use referential nets in a cognitive model of language). Referential nets are related to Discourse Representation Theory (DRT, Kamp & Reyle 1993), but can model all levels of language, particularly the semantic, conceptual and discourse levels. Emphasising the referential nature of language over the truth-values of utterances, they give a better fit to the representations used in UTCs.

On a general level, referential nets fit well to ACT-R's chunks. The main means of structuring referential nets are *referential objects* (*refOs*), which represent the knowledge about an entity, while the main means in ACT-R are

*chunks* representing facts. For example, like chunks, *refOs* can be merged and they can account for inconsistent knowledge, as inferences are only made locally, not globally. More important, both chunks and expressions in referential nets are typed. The main problem for the translation is that referential nets combine these expressions in a principled manner, which is difficult to realise in ACT-R.

Chunks in ACT-R are lists of slot–value pairs, e.g.:

```
(fact3+4 ISA addition-fact
  addend1 three
  addend2 four
  sum seven)
```

represents the fact that  $3 + 4 = 7$  (Anderson & Lebiere 1998: 23). The corresponding chunk-type is:

```
(chunk-type addition-fact addend1 addend2 sum)
```

Thus, all chunks of type *addition-fact* have three slots: *addend1*, *addend2* and *sum*.

*RefOs* are terms (in the sense of formal logic) and are specified by typed expressions. These expressions are either attributes or designations. An example of a *refO* is:

```
human  ──┐ r1
male    ──┘  ──┐ 'DAVID'
                ──┘  father_of('RUTH')
                ──┘  ux wife('SARAH', x)
```

*RefO* *r1* is specified by two attributes (written to the left of the *refO* term) and three designations (written to the right). Designations can be names ('DAVID'), functional expressions (*father\_of*('RUTH')) and descriptions (*ux wife*('SARAH', *x*), meaning 'Sarah is the wife of the entity represented by *r1*').

### Translation

Translating the above expressions is straightforward. Since all expressions in referential nets are typed, names, for example, can simply be defined as chunks of the type:

```
(chunk-type name-fact name)
```

'DAVID' then corresponds to the chunk:

```
(name-fact1 ISA name-fact name david)
```

Descriptions, e.g. *ux wife*('SARAH', *x*), require more work. In addition to types, referential nets also have sorts (corresponding to ACT-R's *isa-hierarchy*, omitted here due to space limitations). The basis of descriptions are sorted predicate–argument structures; so, the sort of *wife* is  $\langle person, person \rangle$ . This corresponds to the chunk-type:

```
(chunk-type wife person1 person2)
```

Abstracted versions of descriptions like the one given above can be captured by chunks of the following type:

```
(chunk-type abs-designation op var predicate)
```

where *op* is a description operator (*t*,  $\eta$ , *some-t*, *all-t*), *var* is the abstraction variable and *predicate* refers to a chunk

containing the predicate, here: a *wife* chunk. (Using variable slot names is not standard but possible in ACT-R.)

While this part of the translation is straightforward, two other issues cause problems. The first one is that attributes can have values, which can be lists. For example, the above refO representing David could stand for a scientist belonging to two research groups, represented by the list attribute *member\_of*(['GRP1', 'GRP2']). In order to represent this as chunks, the list must be spilt up into multiple attributes: *member\_of*('GRP1'), *member\_of*('GRP2').

The second and more severe problem is to translate the assignment of expressions to referential objects, e.g. to represent that *human* is an attribute of *r1*. This is problematic, because refOs are not restricted in their number and kind of attributes or designations. Thus, although refOs and chunks fulfil similar roles in organising and structuring the representation (memory), they differ in that chunks are typed expressions, while refOs are simply terms, i.e. they are proxies of entities.

For simplicity, at this point I will discuss only the assignment of attributes to refOs. Assigning designations (which actually can also have attributes) to refOs works analogously. The two ways to realise these assignments in ACT-R make different predictions in the resulting models. The first solution is to define a general chunk-type and distribute the information over multiple chunks:

```
(chunk-type attrib-assign refo attrib)
(att1 ISA attrib-assign refo r1 attrib human)
(att2 ISA attrib-assign refo r1 attrib male)
```

Under this solution, the knowledge about the entity represented by *r1* cannot be accessed at once. Instead multiple chunk retrievals have to be performed until the required information has been found.

The second solution is to store the attributes in a list and have just one chunk per refO, which means that all attributes are available at once:

```
(chunk-type att-assig refo att)
(assign1 ISA att-assig refo r1 att '(human male))
```

The difference between the solutions is substantial, because solution 1 predicts that accessing knowledge about a refO can take considerable time, while solution 2 finishes the comparison within a single ACT-R cycle.

While the ACT-R theory does not permit lists as slot values (and, thus, predicts behaviour according to solution 1), referential nets allow immediate access to all information about a refO (predicting solution 2). An indication that solution 2 is cognitively valid is the model INC (*incremental conceptualiser*, Guhe et al 2003), which produces reliable simulations of humans performing the task of conceptualisation for language production. The processing of concepts is a good example for determining the correct solution because of the predicted processing times: solution 1 predicts longer processing times for bigger refOs: at least one ACT-R cycle per attribute. (See Guhe in press on why refOs are appropriate to represent concepts.) Empirical support for this prediction of solution 1 is lacking: there is no evidence that familiar concepts (i.e. concepts about which much is known) take longer to process.

In ACT-R 6, the theory's current implementation, the function computing the similarity between two chunks has had to be extended to allow for lists. In a prototype im-

plementation I have realised the similarity function as computing the number of shared attributes over the number of all attributes (cf. Guhe in press), e.g.

```
'(human male) x '(human) = 1/2
'(human male) x '(human woman) = 1/3
```

(The computation is in fact more complex: (1) attributes can contain arguments (*memb\_of\_family* ('SMITH')); (2) these arguments can be lists (*member\_of*(['GRP1', 'GRP2'])); (3) designations must be compared, as well.)

## Discussion

In this short paper I have glossed over many more problems, e.g. the fact that arguments of expressions can also be refOs, so that comparing expressions requires comparing the refOs referred to, as well. In conclusion, however, it seems possible to translate referential nets into ACT-R chunks, differing predictions notwithstanding. Since ACT-R 6 provides a framework for implementing simultaneous access to all of a refO's information, it will support more detailed comparisons within working models. A partial reimplementing of INC, for example, will be one future step on this path.

On a final note, the difficulties caused by the limitations of ACT-R's chunks equally affect the schema-based formalism DRT: as schemas are the main means of structuring these representations, they must be realised as chunks. Yet, schemas are also not limited in the number or kind of expressions they can accommodate. Thus, the problems discussed here are not limited to referential nets but seem to indicate general obstacles in transferring CL representations into ACT-R.

## Acknowledgments

Thanks to Mike Schoelles for discussions of ACT-R and to Ellen Bard and Mark Steedman for comments. This research was supported by grant NSF-IIS-0416128.

## Literature

- Anderson, J.R., & Lebiere C. (1998) *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum.
- Budiu, R. and Anderson, J.R. (2004). Interpretation-based processing: a unified theory of semantic sentence comprehension. *Cognitive Science*, 28(1):1–44.
- Guhe, M. (in press) *Incremental conceptualisation for language production*. Mahwah, NJ: Lawrence Erlbaum.
- Guhe, M., Habel, C. & Tschander, L. (2003) Incremental production of preverbal messages with INC. In: *Proceedings of ICCM-5*, 123–128. Bamberg, Germany.
- Habel, C. (1986) *Prinzipien der Referentialität*. Springer.
- Kamp, H. & Reyle, U. (1993). *From discourse to logic*. Dordrecht: Kluwer.
- Landauer, T.K., Foltz, P.F. & Laham, D. (1996). An introduction to latent semantic analysis. *Discourse Processes*, 25(2/3): 259–284.
- Lewis, R.L. (1993). *An architecturally-based theory of human sentence comprehension*. PhD thesis, CMU.
- Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard UP.
- Reiter, E. (1994). Has a consensus NL generation architecture appeared, and is it psycholinguistically plausible? In: *Proceedings of INLGW-7*, 163–170.