The MKR Language

Dr. Richard H. McCullough
Pioneer, CA 95666
rhm@cdepot.net

Abstract

MKR is a user-friendly RI ("Real Intelligence") language which combines the best features of English, UNIX shell, Unicon and CycL. MKR propositions have a terse English-like format which helps a human user focus on essential characteristics and avoid floating abstractions. MKR is a very-high-level knowledge representation language with a rigorous epistemological foundation including context, genus-differentia definitions, ECP hierarchies (knits) and a unique characterization of the changes associated with actions. The MKE program includes an intelligent knowledge assistant (ke) which interacts with the knowledge base, checks for errors, executes user commands, and possesses self knowledge. This paper gives an overview of the MKR language, and describes several MKE/MKR applications: a Genealogy knowledge base; MKR interfaces to Semantic Web knowledge bases; axiomatic concepts in epistemology.

0. introduction

The MKR language and McCullough Knowledge Explorer (MKE) were "born" in 1996. MKR semantics are the same as English semantics. There is no proof theory for MKR .

1. MKR requirements

correct epistemology:
- context
- genus-differentia definition
- ECP hierarchy (knit)
- action

user friendly (English-like):
- phrase list
- statement
- question
- command
- logic

user-friendly (UNIX-shell-like):
- variable
- method
- control structure

intelligent knowledge assistant (ke):
- record, search, change knit
- check for errors in knit
- execute user commands
- customize processing (variable)
- possess self knowledge

Discovering the proper characterization of the changes associated with actions turned out to be the most difficult part of the MKR design. The current characterization using prepositional phrase lists (pplist) was "born" in 1999. It is one of the most significant new features of MKR.

2. English "problems"

In my opinion, English is "too complicated" to be a good KR language. I eliminated these English "problems" when I designed the MKR language.

ambiguity -- use context/view
redundancy -- use characteristic name = value
nominalization -- use action
verb tenses -- use context/time (past,present,future)
inverted word order -- use statement format
for questions with embedded "?"

Examples of the undesirable redundancy in English are subject-verb agreement and plurals. In MKR, the infinitive form of an English verb is used, regardless of the subject. For example

English: I walk. MKR: I do walk done;
he walks. he do walk done;

3. MKR propositions

Basic format

The simplest form of MKR proposition is
at context { sentence };
where context is
space=s,time=t,view=v
and sentence has the form
subject verb object pplist;

s,t specify where,when actions occur, and v names the contextual knowledge unit (knit) -- an entity-characteristic-proposition hierarchy. This ECP hierarchy includes all the propositions which determine the meaning of the sentence.

Characteristic types

The object describes some characteristics of the subject. The verb indicates what type of characteristic. The basic characteristic types and corresponding verbs are

definition	is,isa,iss,isu
alias	is
hierarchy	isa,iss,isu
group	ismem,isalt
relation	rel
part	haspart
attribute	has
action	do
interaction	causes

Many MKR verbs have inverses and iterative forms, not shown here.

The two forms which use pplist are definitions

```
subject is genus of domain with differentia;
```

and actions

```
subject do action = event
   out  action products
   of   action domains
   with action characteristics
   od   action direct objects
   from action initial characteristics
   to   action final characteristics
   done;
```

Actions may also be written as productions

```
product := subject do ... done;
```

Sentence types

MKR has the classical sentence types found in natural languages, plus some "programming" types.

- statement
- question
- command
- conditional
- iteration
- assignment
- production
- group definition (begin ... end)

Group definitions are used for defining hierarchies, relations, and other groups.

Hierarchy definition

This example is a high-level ECP hierarchy.

```
begin hierarchy overview;
existent;
/  entity;
// man;
// knowledge;
// list;
/  characteristic;
// definition;
// part;
// attribute;
///    conscious;
///    space;
///    time;
///    view;
// relation;
// action;
///    exist;
///    identify;
// interaction;
/  proposition;
// context;
// sentence;
///    statement;
///    question;
///    command;
///    conditional;
///    iteration;
///    assignment;
///    production;
///    group definition;
// Existence;         # entity DO exist done;
// Identity;          # entity HAS characteristic;
// Consciousness;     # man has conscious;
// Identification;    # man do identify od entity done;
end hierarchy overview;
# propositions
Existence IS Identity;
Consciousness IS Identification;
existent is existent;
existent HAS characteristic;
existent DO exist done;
man isa entity;
knowledge := man do identify od existent done;
knowledge haspart proposition list;
proposition isc context, sentence;
```

Relation definition

This example is the birth-death relation used in the MKE Genealogy application.

```
r_birth is relation with
   arraymode=gdbm,arraykey="$1",
   label=[pid, sex, birthdate, birthplace, deathdate,
deathplace, deathcause],
```

```
   format=[person:1, sex:2, time:3, space:4, time:5, space:6, cause:7],
   automatic="isu",
   meaning={
   $1 is person with pid=$1;
   #at space=$4,time=$3 {$1 do exist done;};
   #at space=$6,time=$5 {$1 do die with $7 done;};
   $1 has sex=$2,birthdate=$3,birthplace=$4,
        deathdate=$5,deathplace=$6,deathcause=$7;
   };

begin relation r_birth;
Dick McCullough 1936,male,1936,GA/Ft. Benning,NA, NA, NA;
Bob McCullough 1938,male, 1938, GA/Ft. Benning,NA, NA, NA;
   ...
end relation r_birth;
do check od person done;  # number of persons in KB
```

4. some MKE/MKR applications

Genealogy

MKE includes all the relations needed for a Genealogy knowledge base.

aka	address
birth-death	phone (and email)
marriage-divorce	school
child	work
note	

To reduce processing time (which had become a factor in my 1000 person KB), I replaced some of the MKR meaning scripts with Unicon procedures.

MKE generates a GEDCOM file, which is a standard format used by many Genealogy programs. GEDCOM files are organized by

- person
- family (marriage)
- note

Semantic Web

MKE currently interfaces directly with

- Stanford TAP KB (RDF/OWL language)
- OpenCyc KB (CycL language)
- Google
- Amazon

MKE translates between MKR and RDF,OWL,CycL. The variables

```
kbmode      isany mke,tap,cyc;
kblanguage isany mkr,rdf,owl,cycl;
```

specify which knowledge base MKE accesses and which language it uses. The variable

```
kformat isany mkr,rdf,owl,cycl,ho,rel,...;
```

specifies what input language MKE uses.

Epistemology

By using the MKR language, I have developed some significant new insights into Ayn Rand's axiomatic concepts -- Existence, Identity, Consciousness. I have discovered that Rand's concepts can be identified as the names of the propositions which express the facts of reality.

```
Existence ::        entity DO exist done;
Identity ::         entity HAS characteristic;
Consciousness ::  man has conscious;
```

Further, I can see that there is a "missing" fourth axiomatic concept -- Identification.

```
Identification :: characteristic := man do identify od entity done;
```

Now it becomes quite obvious what Rand's equivalence statements mean. The propositions

```
Existence IS Identity;
Consciousness IS Identification;
```

are statements of metaphysical equivalence among the four axiomatic propositions.

5. availability

McCullough Knowledge Explorer (MKE) is a free, open source program, available for Windows, Linux and Mac. It can be downloaded from http://rhm.cdepot.net/.

References

Ayn Rand, 1990. *Introduction to Objectivist Epistemology.* New York, NY.: Meridian.