A short presentation of YAK, a Hybrid Knowledge Representation System

Enrico Franconi
IRST, I-38050 Povo TN, Italy
franconi@irst.it

YAK is a fully implemented hybrid knowledge representation system of the Kl-One family. The current version – "Peace Now! release 1.2b2", November 1991 – embodies the following characteristics:

• The TBox language, with respect to $\mathcal{FL}^-$, allows for primitive concepts and roles, functional roles and role conjunction; note that, differently from all currently developed systems, there is no number restriction. A tractable and complete subsumption procedure is provided [2].
• An object-oriented ABox with an abstraction / classification / propagation realizing procedure [6, 8].
• A Constraint Box which includes: disjointness, implies rules, transitive roles, test operator [1, 7].
• Constraints inference (á la Allen) added to ABox, to exploit set and time reasoning [5].
• A compositional unification-based query language with typed variables and simple connectives.
• A simple mechanism for multiple KBs handling.
• "Viewpoints" according to the *relevant beliefs* theory for multiple agents beliefs representation [4].

YAK is written in pure common-lisp. An optional graphical interface is available for Xerox Medley common-lisp – featuring display and browsing capabilities, Macintosh Allegro common-lisp, KCL, Allegro and Lucid common-lisp under Xwindow – featuring display only.

In the Appendix a simplified version of the syntax of the language and a brief summary of the functional interface are presented. The functional interface description gives an idea of the kind of interaction that a user (either a knowledge engineer or a lisp application) can have with the system.

In the future I plan to work on: time, actions and plans representation and reasoning; simple revision mechanisms for ABox; interaction with DataBases; a unifier for terminological descriptions; a more expressive TBox; a richer graphical interface for Mac and Xwindow.

The YAK system has been developed initially as the KR module of the ALFresco natural language system, a multimodal dialogue prototype for the exploration of italian art history. Now it is a stand-alone, robust, documented and portable piece of software, used within other AI NL projects.

The YAK package is composed by the following deliverables:
(a) Manual (includes installation hints, syntax specifications, functional interface description, and lisp library reference guide) [3];
(b) Tutorial (includes examples and test cases);
(c) the ALFresco knowledge base (the one with painters and frescoes);
(d) documented common-lisp source code;
(e) a 'defsystem' to load and compile the code.

The package is available with free licence for both academic and not academic sites. The support is unofficial; although I will try to fix bugs that are reported to me. Licensed users have access to major revisions of the system.

Appendix

TBox

```
<concept> ::=  anything |
               lisp-object |
               <atomic-concept> |
               (every <role> <concept>) |
               (test <function> [<arg>]) |
               (and <concept>+) |
               (primitive <concept> [<index>])
```

```
<role> ::=  anyrelation |
            <atomic-role> |
            (single-valued <role>) |
            (and <role>+) |
            (primitive <role> [<index>])
```

ABox

```
<individual> ::=  <atomic-individual> |
                  <individual-description>
```

```
 <individual-description> ::=
     (individual  [<concept>]
                  (fills <role> <individual>)*
                  (<set-relation> <individual>)*
                  (<time-relation> <individual>)*)
```

```
<set-relation> ::= subset | not-subset | superset |
    not-superset | disjoint | not-disjoint | element-of |
    not-element-of | has-element | not-has-element
```

```
<time-relation> ::= before | after | during | contains |
    overlaps | overlapped-by | meets | met-by | starts |
    started-by | finishes | finished-by | equals
```

Functional Interface: Tell

```
<tell> ::=  <term-introduction> |
            <individual-introduction> |
            <constraint> |
            (switch-context <viewpoint>)

<term-introduction> ::=
    (defconcept <atomic-concept> <concept>) |
    (defrelation <atomic-role> <role> [transitive])

 <individual-introduction> ::=
     (close-assert  <atomic-individual>
                    <individual-description>)

<constraint> ::=  (defdisjoint <atomic-concept>+) |
                  (defrule <atomic-concept> <concept>)
```

Functional Interface: Ask

```
<ask> ::=  (subs?  <concept> <concept>) |
           (r-subs?  <role> <role>) |
           (contradict?  <concept>) |
           (retrieve (<variable>*)  <query-individual>
                                    [<atomic-individual>]) |
           (conflict?  <individual>) |
           (believe?  <viewpoint>   <query-individual>
                                    [<atomic-individual>])

<query-individual> ::=  <basic-query-individual> |
                        (and <query-individual>+) |
                        (or <query-individual>+) |
                        (not <query-individual>)

<basic-query-individual> ::=
    <individual> |
    (<variable> <basic-query-individual>) |
    (instantiate <basic-query-individual>) |
```

```
(individual  [<concept>]
             (fills <role> <basic-query-individual>)*
             (<set-relation> <basic-query-individual>)*
             (<time-relation> <basic-query-individual>)*)
```

References

[1] Ronald J. Brachman, Deborah L. McGuiness, Peter F. Patel-Schneider, Lori Alperin Resnick, and Alexander Borgida. Living with CLASSIC: When and how to use a KL-ONE-like language. In J. Sowa, editor, *Principles of Semantic Networks*. Morgan Kaufmann, 1991.

[2] R. Cattoni and E. Franconi. Walking through the semantics of frame-based description languages: A case study. In *Proc. of the 5 th International Symposium on Methodologies for Intelligent Systems*, pages 234–241, Knoxville, TN, 1990.

[3] E. Franconi. The YAK manual. Technical Report 9003-01, IRST, Povo TN, Italy, March 1990. Also as: Progetto Finalizzato CNR 'Sist. Informatici e Calcolo Parallelo' report 7/30.

[4] E. Franconi. Extending hybridity within the YAK knowledge representation system. In *Proc. of the 2 nd International Workshop on Terminological Logics*, Schloss Dagstuhl, Germany, 1991.

[5] E. Franconi. Adding constraints inference to ABox reasoning. Technical Report 9205-02, IRST, Povo TN, Italy, May 1992.

[6] E. Franconi, B. Magnini, and O. Stock. Prototypes in a hybrid language with primitive descriptions. *Computer and Mathematics with Applications, special issue: Semantic Networks in Artificial Intelligence*, 23(6-9):543–556, 1991.

[7] R. MacGregor. A deductive pattern matcher. In *Proc. of AAAI-88*, pages 403–408, St. Paul, MN, 1988.

[8] B. Nebel. *Reasoning and Revision in Hybrid Representation Systems*, volume 422 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, Heidelberg, New York, 1990.