FORSCHUNGSSTELLE FÜR
INFORMATIONSWISSENSCHAFT
UND KÜNSTLICHE INTELLIGENZ

Leitung: Prof. Dr. W. v. Hahn

UNIVERSITÄT HAMBURG
Mittelweg 179
2000 Hamburg 13
Tel.: (040) 4123-4529

HAM ANS

Memo ANS-7

DER SYSTEMRAHMEN VON HAM-ANS

Bernhard Nebel

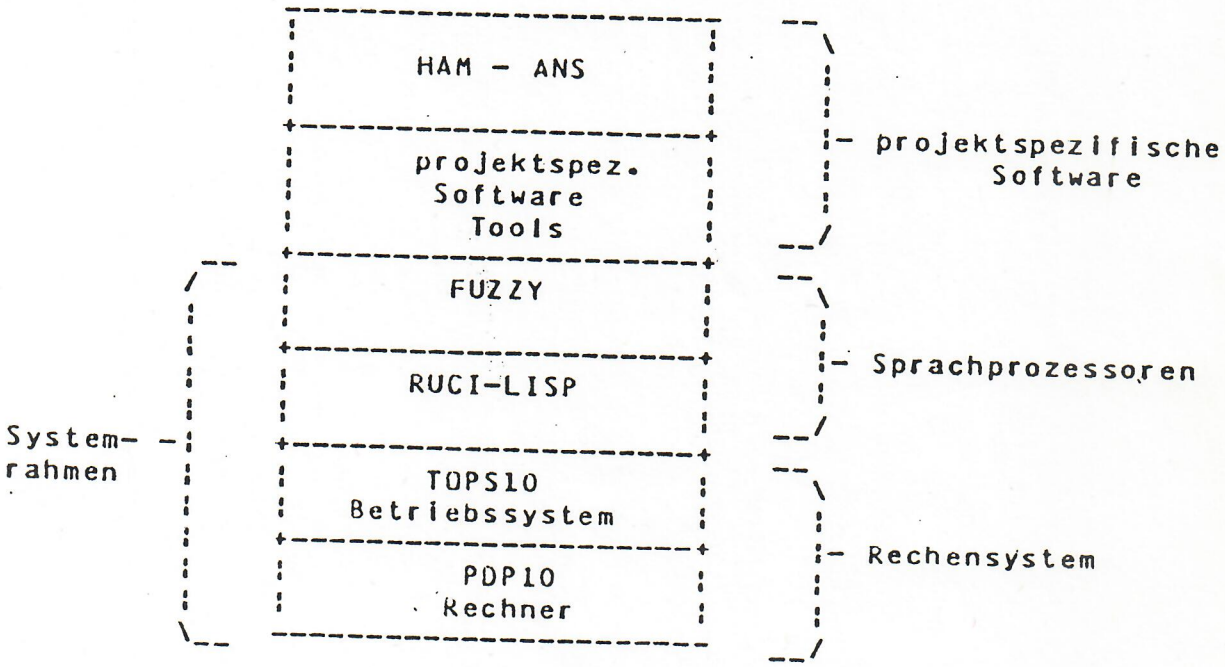
März 1982

Der Systemrahmen von HAM - ANS

Dieses Papier beschreibt, unter welchen Rahmenbedingungen HAM-ANS implementiert ist. Neben einer kurzen technischen Charakterisierung der verwendeten Systeme wird schwerpunktmaessig darauf eingegangen, wie die erforderliche Systemsoftware zu installieren und ggfs. zu generieren ist.

1. Ueberblick ueber das System

Es liegt folgende Schichtung von (mehr oder weniger) abstrakten Maschinen vor:



1.1 Das Rechensystem

HAM-ANS laeuft auf einer Rechenanlage des Fachbereichs Informatik, Universitaet Hamburg. Die Anlage weist folgende Merkmale auf:

- Systemname: DECSystem 1070 (PDP10)
- Prozessor: KI 10
- Wortlaenge: 36 Bit
- Rechenleistung: 0.72 MIPS
- Kernspeicherausbau: 320 KWorte
- Betriebssystem: TOPS10 Version 603A

1.2 Die Sprachprozessoren

Rutgers/UCI-LISP (kurz RUCI-LISP) [LEFAIVRE 78a] ist eine kompatible Erweiterung von UCI-LISP [BOBROW et al 76], welches wiederum aus LISP 1.6 [QUAM & DIFFIE 72] hervorgegangen ist. Eine zusammenfassende Darstellung der Sprache ist in [MEEHAN 79] zu finden. RUCI-LISP ist wie jede andere Lispimplementation nicht nur eine Programmiersprache, sondern auch eine Programmierumgebung, enthaelt also Editor, Prettyprinter, Debug etc.. Zum Starten von RUCI-LISP benoetigt man mindestens 19 KWorte pure Code (sharable High-Segment) und 13 KWorte impure Code (Low-Segment), wenn man die in Hamburg laufende Konfiguration des LISP-Systems benutzt.

FUZZY ist eine in RUCI-LISP eingebettete Planner-artige Sprache, die vage Inferenzen unterstuetzt [LEFAIVRE 78b]. Zum Starten des FUZZY-Systems werden 19KWorte pure Code (das RUCI-LISP High-Segment) und mindestens 24 KWorte fuer das Low-Segment benoetigt.

Von den genannten Interpretern werden folgende Versionen benutzt:

RUCI-Lisp 10K(2)-4 vom 24-MAR-82,
eine gegenueber der von Rutgers gelieferten Version 10K(1)-3 vom 2-JUN-82 leicht modifizierte Fassung (s. Abschnitt 3).

FUZZY vom 24-MAR-82
ist eine gegenueber der von Rutgers gelieferten Version vom 10-MAR-78 modifizierte Fassung (s. auch Abschnitt 3).

1.3 Projektspezifische Software

Die projektspezifische Software setzt sich zusammen aus den Softwarewerkzeugen (ATTACH-Package, DOKU-System etc.), die schon an anderer Stelle beschrieben wurden [HOEPPNER 81] [JAMESON 80], und dem eigentlichen HAM-ANS. Dazu koennen je nach Anwendung noch andere Softwarepakete kommen, z.B. PASCAL/R. Der Speicherbedarf zum Starten von HAM-ANS betraegt 19 KWorte pure Code (RUCI-LISP sharable High-Segment) und mind. 128 KWorte impure Code (Low-Segment).

2. Installation von HAM-ANS

Eine Uebertragung von HAM-ANS auf andere DECSystem-10 oder -20 Rechner mit TOPS10 bzw. TOPS20 sollte ohne grosse Probleme moeglich sein, wenn die Hauptspeicheranforderungen von HAM-ANS erfuellt werden koennen. Zuvor muss jedoch die erforderliche Systemsoftware installiert werden.

Das Vorgaengersystem HAM-RPM wurde mit Erfolg auf verschiedenen DECSystem-10 und -20 Rechnern installiert (siehe [WAHLSTER 79, S.1] und [v.HAHN & WAHLSTER 81]).

2.1 Installation von RUCI-LISP und FUZZY

Zu einem fertig generierten LISP/FUZZY-System gehoeren die folgenden Files:

LISP.EXE	ausfuehrbarer RUCI-LISP Interpreter (das sharable High-Segment dieses Programms wird von LISPC,FUZZY und FUZZYC mitbenutzt)
LISP.LOD	LISP-Loader fuer Maschinenprogramme
LISP.SYM	Symboltabelle von LISP fuer LISP-Loader
LISPC.EXE	der LISP Compiler
FUZZY.EXE	der FUZZY Interpreter
FUZZYC.EXE	der FUZZY Compiler

HAMFUZ.EXE	FUZZY Interpreter mit den HAM-ANS Software Werkzeugen

Ausserdem existieren noch folgende Manual-Files:

SAILSP.MAN	LISP 1.6 Manual
UCILSP.MAN	UCI LISP Manual
RUTLSP.MAN	RUCI LISP Manual
FUZZY.MAN	FUZZY Manual

Je nachdem, in welchem Rahmen LISP und FUZZY benutzt werden sollen, muss noch eine Anpassung vorgenommen werden. Es sind zwei Faelle denkbar:

- Speicherung unter normaler Benutzernummer und Benutzung von verschiedenen Nummern.

- "Offizielle" Benutzung, d.h. Speicherung auf SYS:.

Vorbereitet sind die Programme fuer die zweite Moeglichkeit. Will man den ersten Fall realisieren, so muss man ein neues LISP/FUZZY System generieren und bei der Ausfuehrung von MAKLSP.MIC als Parameter die PPN im LISP Format angeben, unter der LISP.EXE gespeichert werden soll, also z.B.:

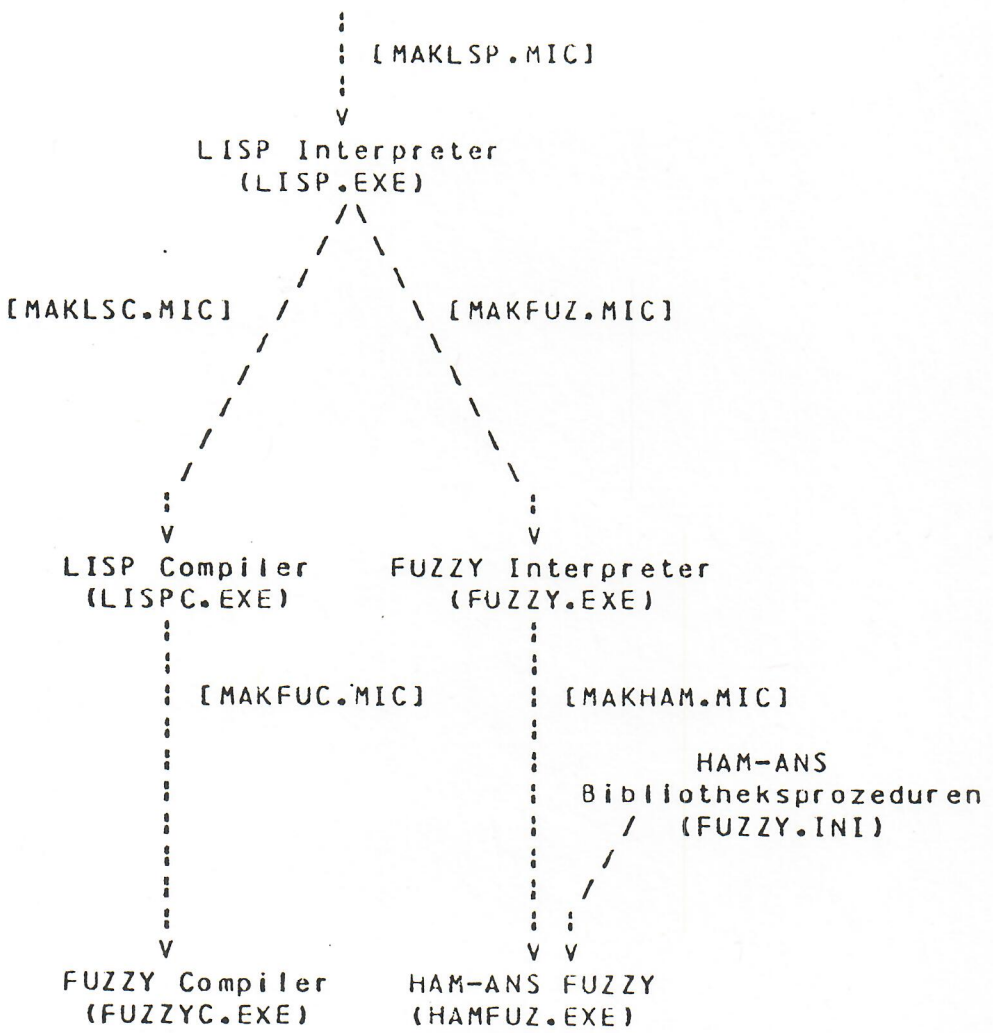
```
DO MAKLSP (316Q 2612Q)
```

Alle MAK???.MIC Files muessen dann unter der PPN ausgefuehrt werden, unter der LISP.EXE gespeichert ist.

2.2 Generierung eines neuen Systems

Sind an LISP und/oder FUZZY Aenderungen vorgenommen worden, oder soll das System umkonfiguriert werden (z.B. wie oben), so ist ein neues System zu generieren. Dafuer sind die MAK???.MIC Kommandofiles vorgesehen. Als Parameter ist bei MAKLSP eine PPN im LISP Format (s.o.) oder SYS: (optional) zulaessig. Bei allen anderen Files kann man DSK: oder SYS: (optional) angeben. Daraufhin wird LISP von SYS: oder dem aktuellen Bereich gestartet (das ganze ist eine etwas undurchsichtige Angelegenheit, da nach dem Start durchs RUN Kommando das High-Segment geladen wird, dass durch SETSYS bestimmt wurde).

Bei der Generierung sind die folgenden Abhaengigkeiten zu beachten (die Pfeile geben die zeitliche Reihenfolge an, in der die Kommandofiles ausgefuehrt werden muessen):



In runden Klammern steht jeweils das ausfuehrbare Programm, in eckigen der Kommandofile zum Erzeugen dieses Programms.

Im folgenden sind alle Programme, Module und Kommandofiles aufgefuehrt, die fuer die Erzeugung der einzelnen ausfuehrbaren Files (EXE-Files) benoetigt werden. Die Quellcodefiles tragen die Extension ".MAC" (fuer MACRO-10 Assembler) oder ".LSP" (fuer LISP). Die Objektcodefiles haben die Extension ".REL" (relozierbarer Objektcode) oder ".LAP" (LISP Assembler Code). Kommandofiles, die mit DO aufgerufen werden, tragen die Extension ".MIC"; welche, die innerhalb des LISP Interpreters eingelesen werden, haben die

Extension ".BLD". Bei einer Aenderung an den Quellfiles ist vor einer neuen Generierung natuerlich eine Uebersetzung durchzufuehren.

2.2.1 Generierung des LISP Interpreters

Programme & Module

LISP.MAC	& .REL	der in MACRO-10 geschriebene Teil des LISP Kerns
LISP.LSP	& .LAP	der in LISP geschriebene Teil des LISP Kerns
BREAK.LSP	& .LAP	BREAK-Package
ERRORX.LSP	& .LAP	ERROR-Package
EDIT.LSP	& .LAP	der LISP Editor
PP.LSP	& .LAP	der Prettyprinter
LAP		der LISP "Assembler", dient zum Einlesen von compilierten Programmen.
SYMMAK.MAC	& .REL	dient zum Erzeugen der Symboltabelle fuer den LISP LOADER
LOADER.MAC	& .REL	alter DEC LOADER. Dient zum erzeugen des LISP LOADERs

Kommandofiles

MAKLSP.MIC	erzeugt kompletten Interpreter
LISP.BLD	liest alle Module (BREAK, PP etc.) ein
LISP.CMD	fuer den Assembler und Linker beim Uebersetzen von LISP.MAC resp. LISP.REL
SYMMAK.CMD	dito fuer SYMMAK.MAC und .REL

erzeugte Files

LISP.EXE	der fertige Interpreter
LISP.LOD	der LISP LOADER. Wird beim Aufruf der Funktion LOAD geladen
LISP.SYM	Symboltabelle vom LISP fuer den LOADER
REMOB.LSP	alle von der OBLIST entfernten Symbole
LISP.REL	wenn nicht schon vorhanden
SYMMAK.REL	- " -
LOADER.REL	- " -

2.2.2 Generierung des LISP Compilers

Der LISP Compiler wird nicht als zusaetzliches Paket in den Interpreter eingeladen, da seine interaktive Benutzung nicht unbedingt erforderlich ist. Stattdessen existiert ein separates Programm, das allerdings das "sharable High Segment" des LISP Interpreters mitbenutzt. D.h. nach einer Modifikation am Interpreter muss auch der Compiler neu generiert werden (s.a. das Abhaengigkeitsdiagramm oben).

Programme & Module

LISP.EXE	der LISP Interpreter. Das Compiler Modul wird erst ins Low-Segment eingelesen, danach wird dieses mit einem Verweis auf das sharable High-Segment des LISP Interpreters (SETSYS) geSAVEt.
LISPC.LSP & .LAP	der LISP Compiler

Kommandofiles

MAKLSC.MIC	Startet LISP.EXE, liest LISPC.BLD ein und SAVEt LISPC
LISPC.BLD	liest LISPC.LAP ein und setzt INITFN

erzeugte Files

LISPC.EXE	der ausfuehrbare LISP Compiler

2.2.3 Generierung des FUZZY Interpreters

Programme & Module

LISP.EXE	wie unter 2.2.2
FUZZY2.LSP & .LAP	FUZZY Interpreter

Kommandofiles

MAKFUZ.MIC	erzeugt FUZZY.EXE
FUZZY.BLD	

erzeugte Files

FUZZY.EXE	FUZZY Interpreter

2.2.4 Generierung des FUZZY Compilers

Programme & Module

LISPC.EXE	LISP Compiler
FUZZY.LSP & .LSP	FUZZY

Kommandofiles

MAKFUC.MIC
FUZZYC.BLD

erzeugte Files

FUZZYC.EXE	der ausfuehrbare FUZZY Compiler

2.2.5 Generierung einer FUZZY Version fuer HAM-ANS

Um den Start von HAM-ANS zu beschleunigen, wird ein speziell auf HAM-ANS zugeschnittenes FUZZY erzeugt, das alle Bibliotheksprozeduren etc. schon eingelesen hat. Dieses wird erreicht, indem FUZZY einmal gestartet und FUZZY.INI eingelesen wird.

Programme & Module

FUZZY.EXE	diesmal wird nicht LISP sondern FUZZY "erweitert"
FUZZY.INI	der Initialisierungsfile fuer das HAM-ANS FUZZY
div Bibliotheks-proz.	siehe [HOEPPNER 81]

Kommandofiles

MAKHAM.MIC	hier sind als Parameter zulaessig: 1. fuer LISP HISEG: DSK:,SYS: oder nichts 2. optionale PPN, wo FUZZY zu finden ist 3. optionaler neuer Name
HAMFUZ.BLD	

erzeugte Files

HAMFUZ.EXE	die HAM-ANS FUZZY Version

3. Modifikationen an der Systemsoftware

Im folgenden wird beschrieben, welche Aenderungen an LISP und FUZZY vorgenommen wurden und welche Auswirkungen das hat.

3.1 Aenderungen am RUCI-LISP

Der Start des High-Segments von LISP wurde von 400000 (oktal) auf 700000 (oktal) verlegt, so dass fuer impure Code im Low-Segment ein potentieller Adressraum von 224 KWorten statt vorher 128 KWorten zur Verfuegung steht. Dieser Schritt wurde notwendig, da HAM-ANS die urspruenglichen 128 KWorte fast vollstaendig belegte.

Ein Effekt bei der Verlegung des High-Segments ist, dass der INUM Bereich von 2^16-1..-2^16 auf 2^14-1.. -2^14 zusammenschrumpft. Damit aendert sich auch die Repraesentation von INUM0, der internen Darstellung der 0 im INUM Bereich. Da der LISP Compiler bei der Uebersetzung von LEXPRs sich auf die Darstellung der INUM0 (zur Compilezeit) bezieht, gibt es bei der Ausfuehrung von LSUBRs, die mit dem alten LISP Compiler erzeugt wurden, unter dem neuen LISP Interpreter Fehler (ILL MEM REFs). Aufgetreten ist dieser Fall bei den Standardfunktionen SOME und EVERY, der File LISP.LAP musste also neu erzeugt werden.

Bei der Benutzung des neuen LISP mit einem CORE Argument beim RUN Kommando, das kleiner als 128 KWorte ist, ergeben sich im Verhalten gegenueber dem alten System keine Aenderungen. Allgemein sollte man folgendes beachten: Bei einem groesseren (virtuellen) Adressraum laeuft das Programm laenger ohne vom "Grabage Collector" (GC) unterbrochen zu werden, d.h. i.a. wird das Verhaeltnis von Programmlaufzeit zu Garbage Collection Laufzeit besser, aber

- wenn der GC dann zuschlaegt, dauert es laenger;
- falls der physikalisch verfuegbare Speicher sehr klein gegenueber dem virtuellen ist, steigt die "Paging Rate" an, und damit die Ausfuehrungs- und Antwortzeit.

3.2 Modifikationen an FUZZY

Zur Adaption von FUZZY an das ATTACH Package, insbesondere zur Speicherung von assoziativen Netzen auf sekundaeren Speichermedien, waren Aenderungen im FUZZY Interpreter (Version vom 10-MAR-78) erforderlich (siehe [JAMESON 80]). Es handelt sich dabei um einige CALL-Definitionen, um die so definierten Funktionen (ZGETAS,ZADD,ZREMOVE,ZERROR) redifinierbar zu machen.

Die spaetere Version von FUZZY vom 8-JUN-78 kann nicht benutzt werden, da die Verwaltung der assoziativen Netze so weitgehend geaendert worden ist, dass an eine Anpassung an das ATTACH Package nicht zu denken war.

Die Arbeiten an HAM-ANS werden aus Mitteln des Bundesministeriums für Forschung und Technologie gefördert.

Literatur

BOBROW et al 76 Bobrow,R.J., Burton,R.R, Jacob,J.M., Lewis,D.
UCI LISP Manual
Dept. of Information and Computer Science,
Technical Report 21
University of California, Irvine, Oct. 1973

HOEPPNER 81 Hoeppner,W.
Eine Uebersicht zur FUZZY-Programmierumgebung
und den Dateizugriffen in HAM-ANS
Forschungsstelle fuer Informationswissenschaft
und Kuenstliche Intelligenz, Memo ANS-1
Universitaet Hamburg, Sep. 1981

JAMESON 80 Jameson,A.
ATTACH: A Package for Accessing LISP Programs
and Data from Disc
Projektgruppe Simulation von Sprachverstehen,
Memo RPM-12
Univ. Hamburg, Maerz 1980

LEFAIVRE 78a LeFaivre, R.
Rutgers/UCI LISP Manual.
Computer Science Dept.
Rutgers Univ., May 1978

LEFAIVRE 78b LeFaivre, R.
FUZZY Reference Manual
Computer Science Dept.
Rutgers Univ., June 1978

MEEHAN 79 Meehan,J.
The new UCI-LISP Manual
Lawrence Erlbaum, Hillsdale 1979

QUAM & DIFFIE 72
Quam,L.H., Diffie,W.
Stanford LISP 1.6 Manual
Stanford AI Lab., Operation Note 28.7
Stanford Univ., 1972

WAHLSTER 79 Wahlster,W.
Ein Ueberblick zur Systemsoftware fuer HAM-
RPM
Projektgruppe Simulation von Sprachverstehen,
Memo RPM-8
Universitaet Hamburg, Sep. 1979

v.HAHN & WAHLSTER 81
v.Hahn,W., Wahlster,W.
Bericht ueber eine Vortrags- und
Informationsreise zu Zentren
sprachorientierter KI-Forschung in den USA
Projektgruppe Simulation von Sprachverstehen
Universitaet Hamburg, 1981

RESEARCH UNIT FOR INFORMATION SCIENCE AND ARTIFICIAL INTELLIGENCE

REPORTS

RPM-1 v. Hahn, Walther, Henskes, Dieter, Hoeppner, Wolfgang, Wahlster, Wolfgang: HAM-RPM: Ein Redepartnermodell als Simulationsprogramm. In: Braunmueller, K., Kuerschner, W. (eds.): Grammatik. Akten des 10. Linguistischen Kolloquiums Tuebingen 1975. Band II. Tuebingen: Niemeyer 1976. 337-357. #

RPM-2 Wahlster, Wolfgang, v. Hahn, Walther: Einige Erweiterungen des natuerlichsprachlichen AI-Systems HAM-RPM. In: Laubsch, J., Schneider, H-J. (eds.): Dialoge in natuerlicher Sprache und Darstellung von Wissen. Arbeitstagung der Fachgruppe 'Kuenstliche Intelligenz' der Gesellschaft fuer Informatik. Freudenstadt 1976. 204-225. #

RPM-3 v. Hahn, Walther, Hoeppner, Wolfgang, Jameson, Anthony, Wahlster, Wolfgang: HAM-RPM: Natural Dialogues with an Artificial Partner. In: Proceedings of the AISB/GI Conference on Artificial Intelligence. Hamburg 1978. 122-131. #

RPM-4 v. Hahn, Walther: Ueberlegungen zum kommunikativen Status und der Testbarkeit von natuerlichsprachlichen Artificial-Intelligence-Systemen. In: Sprache und Datenverarbeitung 1, 1978, 145-169.

RPM-5 Wahlster, Wolfgang: Die Simulation vager Inferenzen auf unscharfem Wissen: Eine Anwendung der mehrwertigen Programmiersprache FUZZY. In: Ueckert, H., Rhenius, D. (eds.): Komplexe menschliche Informationsverarbeitung. Bern: Huber 1979. 249-259. #

RPM-6 v. Hahn, Walther: Probleme der Simulationstheorie und Fragepragmatik bei der Simulation natuerlichsprachlicher Dialoge. In: Ueckert, H., Rhenius, D. (eds.): Komplexe menschliche Informationsverarbeitung. Bern: Huber 1979. 260-269. #

RPM-7 Wahlster, Wolfgang, Jameson, Anthony, Hoeppner, Wolfgang: Glancing, Referring and Explaining in the Dialogue System HAM-RPM. In: American Journal of Computational Linguistics, Microfiche 77, 1978. 53-67.

RPM-8 v. Hahn, Walther: Ueber Dialogkohaerenz in natuerlichsprachlichen AI-Systemen. Oktober 1979. (English version: Coherency of Dialogues in Natural Language AI-Systems)

RPM-9 Wahlster, Wolfgang: Algorithmen zur Beantwortung von 'Warum'-Fragen in Dialogsystemen. Januar 1979. To appear in: Krallmann, D. (ed.): Theorie der Frage. Tuebingen: Narr 1980. #

RPM-10 v. Hahn, Walther: Ueberlegungen zum Handlungsrahmen von Fragen in Artificial-Intelligence-Systemen. Januar 1979. To appear in: Krallmann, D. (ed.): Theorie der Frage. Tuebingen: Narr 1980.

RPM-11 Wahlster, Wolfgang: ATNs und die semantisch-pragmatische Steuerung der Analyse und Generierung natuerlicher Sprache. In: Christaller, Th., Metzing, D. (eds.): Augmented Transition Network Grammatiken. Aspekte des ATN-Formalismus in sprachverarbeitenden Systemen. Berlin: Einhorn 1979. 167-185.

RPM-12 v. Hahn, Walther, Hoeppner, Wolfgang, Jameson, Anthony, Wahlster, Wolfgang: The Anatomy of the Natural Language Dialogue System HAM-RPM. Mai 1979. In: Bolc, L. (ed.): Natural Language Based Computer Systems. Muenchen: Hanser/Macmillan 1980, 119-253. #

RPM-13 Hoeppner, Wolfgang, Jameson, Anthony: Kooperatives Dialogverhalten im Simulationssystem HAM-RPM. In: Proceedings of the 4th Workshop on Artificial Intelligence, Bad Honnef, 1979. 21-31.

RPM-14 Wahlster, Wolfgang: Implementing Fuzziness in Dialogue Systems. November 1979. To appear in: Rieger, B. (ed.): Empirical Semantics. Bochum: Brockmeyer 1980. #

RPM-15 Hoeppner, Wolfgang: Repraesentationsstrukturen und Inferenzen fuer zusammengesetzte Objekte. Maerz 1980. In: Rollinger, C.-R., Schneider, H.-J. (eds.): Inferenzen in natuerlichsprachlichen Systemen der Kuenstlichen Intelligenz. Berlin: Einhorn 1980. 151-171.

RPM-16 Wahlster, Wolfgang: Towards a Computational Model for the Semantics of Why-questions. In: Proceedings of the 8th International Conference on Computational Linguistics (COLING80), Tokio, 1980. 144-150. #

RPM-17 Jameson, Anthony, Hoeppner, Wolfgang, Wahlster, Wolfgang: The Natural Language System HAM-RPM as a Hotel Manager: Some Representational Prerequisites. In: R. Wilhelm (ed.): GI-10. Jahrestagung Saarbruecken. Berlin: Springer 1980. 459-473.

RPM-18 Wahlster, Wolfgang: Natuerlichsprachliche KI-Systeme: Entwicklungsstand und Forschungsperspektive. In: Siekmann, J. (ed.): GWAI-81. German Workshop on Artificial Intelligence. Bad Honnef, January 1981. (Informatik-Fachberichte Bd. 47), Berlin, Heidelberg, N.Y.: Springer 1981. 50-68.

(# ::= Report out of print)

RESEARCH UNIT FOR INFORMATION SCIENCE AND ARTIFICIAL INTELLIGENCE

REPORTS

GEN-1 Wahlster, Wolfgang: KI-Verfahren zur Unterstuetzung der aerztlichen Urteilsbildung. In: W. Brauer (ed.): GI-11. Jahrestagung. Berlin: Springer 1981, 568-579

ANS-2 Jameson, Anthony, Wahlster, Wolfgang: User Modelling in Anaphora Generation: Ellipsis and Definite Descriptions. Dezember 1981. To appear in: Proceedings of the First European Conference on Artificial Intelligence. Orsay 1982

ANS-3 Hoeppner, Wolfgang: A Multilayered Approach to the Handling of Word Formation. March 1982. To appear in: J. Horecky (ed.): COLING-82 - The Ninth International Conference on Computational Linguistics, Prague. Amsterdam: North Holland 1982

MEMOS

RPM-0 Jameson, Anthony: Eine Einfuehrung in die interaktive Arbeit und die Programmierung mit FUZZY. 1978. #

RPM-1 Wahlster, Wolfgang: Eine kurze Einfuehrung in die Organisation der assoziativen Netze in der Programmiersprache FUZZY. Februar 1978. #

RPM-2 Hoeppner, Wolfgang: Nominalgruppen mit unbestimmtem Artikel. Februar 1978. #

RPM-3 Wahlster, Wolfgang: Zur Ueberfuehrung eines Konstituenten-strukturbaums in eine semantische Repraesentations-konstruktion mit Hilfe von Pattern-Matching-Operationen. Mai 1978. #

RPM-4 Hoeppner, Wolfgang, Wahlster, Wolfgang: Die Performanz von HAM-RPM: Zwei kommentierte Beispieldialoge und ein Interaktionsbeispiel. Juni 1978. #

RPM-5 Hoeppner, Wolfgang: NP-Referenzanalyse in HAM-RPM: Kommentierte Performanzbeispiele. Juli 1978. #

RPM-6 Demnick, Gudrun, Marburger, Heinz: Benutzeranleitung fuer das System HAM-RPM. Februar 1979.

RPM-7 Wahlster, Wolfgang: MASTER INDEX: From LISP to AIMDS. August 1979.

RPM-8 Wahlster, Wolfgang: Ein Ueberblick zur Systemsoftware fuer HAM-RPM. September 1979.

RPM-9 Wahlster, Wolfgang: Zur Verarbeitung von Sonderzeichen und Satzzeichen in HAM-RPM. Oktober 1979.

RPM-10 Hoeppner, Wolfgang: Analyse ueber Satzpatterns: Performanzbeispiele fuer WELCH- und WIEVIEL- Fragen. Dezember 1979.

RPM-11 Hoeppner, Wolfgang, Wahlster, Wolfgang: Dialogsequenzen mit dem System HAM-RPM im Kommentierungsmodus. Januar 1980. #

RPM-12 Jameson, Anthony: ATTACH: a Package for accessing LISP Programs and Data from Disk. Maerz 1980.

ANS-1 Hoeppner, Wolfgang: Eine Uebersicht zur FUZZY-Programmierumgebung und den Dateizugriffen in HAM-ANS. September 1981.

GEN-2 Wahlster, Wolfgang, v. Hahn, Walther: Mensch-Maschine-Kommunikation auf der Basis natuerlicher Sprache. Positionspapier zur Arbeitstagung 'Mensch-Machine-Schnittstelle', Univ. Stuttgart 12.-13. 10. 81. Oktober 1981.

ANS-3 Gnefkow, Wilhelm: Studien zu einer Programmierumgebung fuer Augmented Transition Networks (ATN). Januar 1982.

ANS-4 Jameson, Anthony: Documentation for three HAM-ANS Components: Ellipsis, NORMALIZE and NORMALIZE-1. November 1981.

GEN-5 Hussmann, Michael, Genzmann, Heinz: Performanz-orientiertes Parsing - Ansaetze zur robusten Analyse natuerlicher Sprache. Februar 1982.

ANS-6 Wender, Herbert: Dokumentationsprinzipien im Projekt HAM-ANS. Maerz 1982

(# ::= Memo out of print)