

Expertensysteme ohne Geheimnisse

Eine Einführung in die Entwicklungsumgebung *MakeIt*

Marco C. Bettoni,
Projektleiter Künstliche Intelligenz
Fachhochschule beider Basel, m.bettoni@fhbb.ch

Einleitung

Wie das erste Auftreten *menschlicher Intelligenz*¹ aus der Formung ganz einfacher Stein- und Knochen-Artefakte wie Faustkeile, Stichel, Harpunen, usw. erkennen lässt, so ist das erste Auftreten *künstlicher Intelligenz* (ab ca. 1955 bis heute) ebenfalls nur durch ganz einfache "Software-Artefakte" gekennzeichnet: Das Gebiet der Künstli-

hingegen viel, u.a. dass sie häufig äusserst komplex sind, manchmal gar geheimnisvoll.

Einfache KI-Artefakte und komplexe KI-Technologien: Heisst dies, dass für uns heute, insbesondere aus unternehmerischer und ingenieurmässiger Sicht, KI nicht interessant ist? Das hier besprochene Software-System *MakeIt* - eine Entwicklungsumgebung - will ihnen die Möglichkeit geben, dieser Frage am Beispiel der Expertensysteme selbständig nachzugehen und zwar durch "Ausprobieren", ohne Theorie und ohne vorangehende KI-Kenntnisse.

Was sind XPS ?

Wir haben für *MakeIt* das Gebiet der Expertensysteme (abgekürzt XPS) gewählt, erstens weil die XPS-Technologie einen massgeblichen

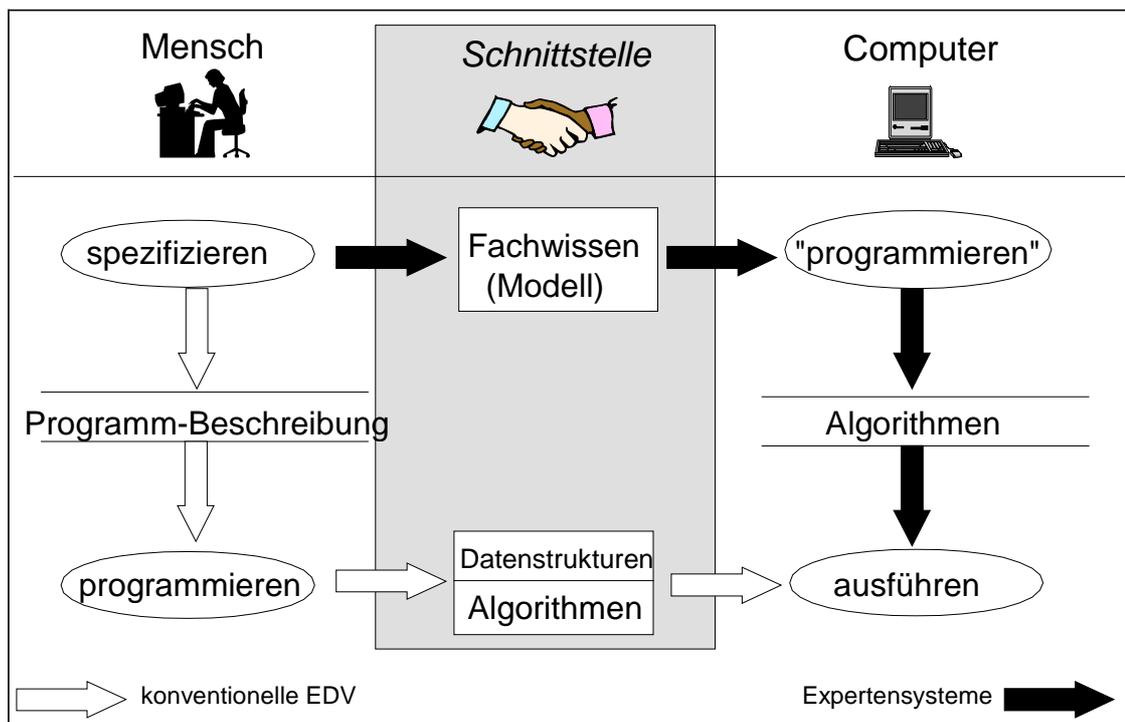


Bild 1: Aufgabenteilung zwischen Mensch und Computer.

chen Intelligenz (KI) ist ein Teil der Informatik und befindet sich sozusagen in seiner "Altsteinzeit". Von den Technologien der Menschen im Paläolithikum, der Prähominiden, wissen wir leider wenig, von den modernen KI-Technologien (z.B. Expertensysteme)

¹ im Paläolithikum, ab ca. 600'000 v. Chr.

Einfluss auf die moderne SW-Technologien (4GL-Sprachen, Datenbanken, usw.) hat und haben wird², zweitens weil XPS jener Teilbereich der KI sind, der

² Ian Graham, Object Oriented Methods, Addison-Wesley, 1994, p.379.

bisher am erfolgreichsten gewesen ist³ und drittens weil wir am CZM die XPS-Technologie sowohl bei der Entwicklung von Applikationen einsetzen (z.B. MASTER) als auch im Nachdiplomstudium (NDS1 und NDS2) unterrichten.

Die XPS-Technologie lässt sich am besten durch Vergleich mit der konventionellen Software-Technologie charakterisieren. Nehmen wir z.B. die *Aufgabenteilung zwischen Mensch und Computer* (Bild 1). Bei der konventionellen SW-Entwicklung

Anders bei der XPS-Technologie. Hier spezifiziert der Mensch sein Wissen (*know-how*) über das Anwendungsgebiet. Wichtig ist, dass er sich dabei nicht um die Umsetzung dieses Fachwissens in Computeralgorithmen zu kümmern braucht: er kann sich voll auf sein Fachgebiet und auf die Problemlösung konzentrieren, was möglich ist, weil hier der Computer über Mitteln verfügt (Inferenzmaschine) die es ihm ermöglichen, ausgehend vom spezifizierten Wissen (Wissensbasis) zur Laufzeit

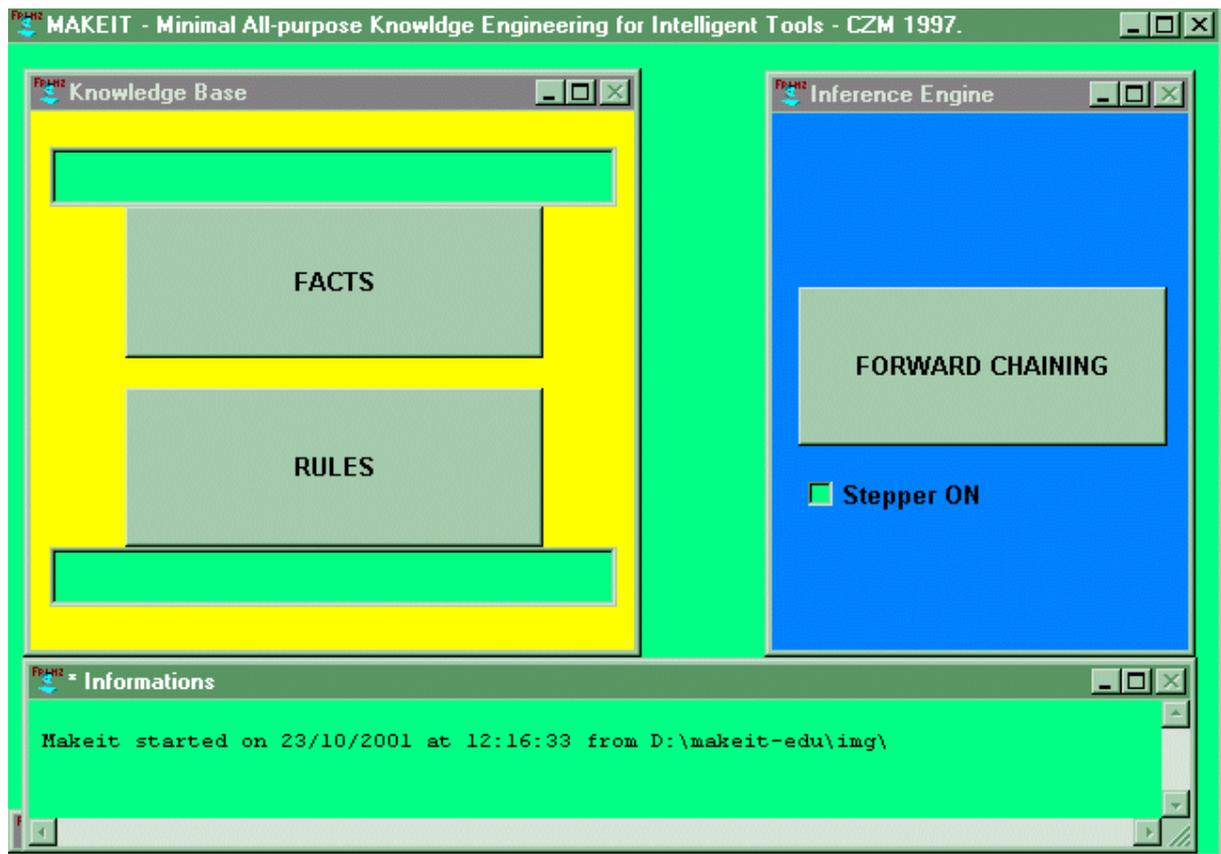


Bild 2: Hauptfenster von *MakeIt*.

spezifiziert der Mensch was die Anwendung tun soll und wie sie, um dies tun zu können, aufgebaut sein muss: Es entsteht eine "Programm-Beschreibung". Danach muss der Entwickler diese Spezifikation durch Programmieren in ablauffähige Algorithmen umsetzen. Dem Computer bleibt am Ende nichts anderes zu tun, als die vorgegebenen Algorithmen auszuführen.

die notwendigen *Algorithmen zu erzeugen*: Das Ausführen der Algorithmen geschieht in Expertensystemen sozusagen im Takt mit ihrer Erzeugung.

Was ist *MakeIt* ?

Soweit die Theorie. Unsere Erfahrung mit dem Unterricht in Expertensystemen hat nun gezeigt, dass theoretische Erklärungen sowie Selbststudium, Übungen auf dem Papier, die Bedienung von XPS-Anwendungen und anderes mehr nicht genügen, um den Unterschied zwischen konventioneller SW-Technologie und XPS-Technologie soweit zu verstehen, dass keine *unerwünschten Geheimnisse*

³ Von 1. Mia US\$ die 1993 das Volumen des weltweiten KI-Marktes bildete, entfielen 595 Mio auf Expertensysteme (Quelle: IEEE-Expert, April 1994).

bestehen bleiben. Unsere Studenten waren jeweils mit dem erreichten Stand des Wissens sehr zufrieden, äusserten aber immer den Wunsch, selber "Hand anzulegen" um sozusagen "hinter die Kulissen" blicken zu können: d.h., sie wollten am Computer selber Expertensysteme entwickeln.

Versuche mit den besten kommerziellen Entwicklungsumgebungen für XPS haben indessen gezeigt, dass diese im Rahmen der zur Verfügung stehenden Zeit nicht brauchbar sind, da sie zu viel Aufwand für das Erlernen der Bedienung erforderten. Paradoxerweise liegt der Grund gerade in der Stärke dieser Produkte (Mächtigkeit und Vielfalt der Bedienmöglichkeiten) !

Der berechtigte Wunsch der Studenten und die Unmöglichkeit, diesen mit kommerzieller Software zu befriedigen führten zum Entschluss, am CZM ein eigenes Produkt zu entwickeln.

MakeIt wurde für Windows '95 konzipiert und am PC mit der KI-Sprache *Lisp* entwickelt. Der Name ist ein Akronym für "Minimal All-purpose Knowledge Engineering for Intelligent Tools" und charakterisiert damit die wesentlichen Eigenschaften des Systems. Seinen Anwendern und Anwenderinnen ermöglicht *MakeIt* Expertensysteme ("Intelligent Tools") für beliebige Fachgebiete ("All-purpose") zu konstruieren ("Knowledge Engineering") indem es einen auf das Minimum reduzierten Satz von Bedienfunktionen anbietet ("Minimal").

Aufbau und Bedienung

Bild 2 zeigt das Hauptfenster von *MakeIt*. Dargestellt sind die Komponenten des Kerns eines jeden Expertensystems, nämlich Fakten (facts) und Regeln (rules) welche zusammen auch Wissensbasis (Knowledge Base) genannt werden, sowie Vorwärts-Verkettung (Forward Chaining) als einzige Funktionsweise der Inferenzmaschine (Inference Engine).

Die Bedienmöglichkeiten sind minimal und das Konzept sehr einfach (*direct manipulation*): die 3 Komponenten sind zugleich die 3 wichtigsten Bedienungselemente.

Für einige weiteren Funktionen ruft *MakeIt* das bekannte NOTEPAD Editor (Windows Zubehör) auf.

In der *Entwicklungsphase* arbeiten die Anwender und Anwenderinnen nur mit Fakten und Regeln (die Inferenzmaschine kann nicht verändert werden): Durch Klicken auf die Komponente 'Facts' kann die Liste der Fakten editiert werden während das Klicken auf die Komponente 'Rules' es ermöglicht, entweder die Liste der Regeln zu editieren oder sie analysieren zu lassen. Beim Formulieren von Fakten und Regeln folgen die Anwender dem einfachsten objektorientierten Prinzip: die Unterscheidung von Objekt, Attribut und Wert welche mit Hilfe von

*Triplen*⁴ realisiert wird.

In der *Anwendungsphase* kommt nur die Inferenzmaschine zum Zug: durch Klicken auf die Komponente 'Forward Chaining' erstellt sie die für die Verarbeitung notwendigen Algorithmen und führt sie aus. Ein *Trace* (Spur) dieses Vorgangs wird im Laufe der Verarbeitung am 'Notification Window' (Bild 2, unten) ausgegeben und ermöglicht es, die Arbeitsweise der Inferenzmaschine zu rekonstruieren sowie nachzuvollziehen. Das Ergebnis, d.h. die hergeleiteten Fakten, wird am Ende zusammen mit den vorgegebenen Fakten als Liste angezeigt (via NOTEPAD).

Installation

MakeIt wird auf einer Diskette in komprimierter Form geliefert⁵ und erfordert 3 MB freien Speicherraum. Ein einfaches *Setup*-Programm hilft Ihnen bei der Installation, ein *Readme* File beschreibt Bedienung, Bestandteile und Grundbegriffe, eine kleine *Beispielsammlung* zeigt mögliche Anwendungsformen und ein *Tutorial* mit gelösten Aufgaben führt Sie Schritt für Schritt in die Entwicklung von Expertensystemen ein.

Möchten Sie die Geheimnisse der Expertensystem-Technologie lüften, hinter die Kulissen der Künstlichen Intelligenz einen Blick werfen? *Do it with MakeIt !*

⁴ Formalisierung von Wissen (Autofahren) mit *Triplen* in *MakeIt*:

A) Wissen:

Regel: Wenn die Ampel auf grün wechselt dann Motor starten und Gas geben

Faktum: Die Ampel hat auf grün gewechselt

B) Formalisiertes Wissen in *MakeIt*:

Objekte: Ampel, Fahrer

Attribute: Farbe, Aktion

Regel: IF (Farbe Ampel grün)

THEN

(Aktion Fahrer "Motor starten")

(Aktion Fahrer "Gas geben")

Faktum: (Farbe Ampel grün)

⁵ Senden Sie eine leere, formatierte Diskette an Marco C. Bettoni, CIM-Zentrum Muttenz, St.Jakobs-Str. 84, 4132 Muttenz.