Aufgaben-Liste
Wissens-Techniken

Wissens-Techniken, Grundlegendes:

1. Wie sind Wissens-Techniken abgegrenzt vom Wissens-Pioneering?
Wissenspioneering bezieht sich auf die Entwicklung von neuem, bisher noch nicht dagewesenem Wissen. Wissenstechniken unterstützen das Wissenspioneering dahingehend, dass im „luftleeren“ Raum bestimmte Techniken angewendet werden können, die die Eigenkonstruktionen unterstützen.
(Deine Quelle war irrelevant, deshalb habe ich sie gelöscht.)
Siehe Folie 5...
Der Wissenspionier ist der HACKER oder SPIELER, der seine Entdeckungen in einer RISKANTEN, aber auch INTERESSANTEN und VIELVERSPRECHENDEN UMGEBUNG macht; aber ohne NETZ bzw. SPIEL kann er wenig anfangen – bei den Wissens-Techniken hingegen wird INFRASTRUKTUR GESCHAFFEN: Z.B. Programme (auch ‚hackbare‘ Netzstrukturen oder Spiele).

2. Wie sind Wissens-Techniken abgegrenzt von der Wissens-Politik?
Wissenspolitik bezieht sich auf das „wie“ der Umsetzung von Wissensmanagement im Unternehmen und darum, wie bestehendes Wissen gespeichert werden kann. Wissenstechnik dagegen um das „wie“ der Generierung neuen Wissens.
Siehe Folie 6...
Der Wissenspolitiker ist der HIRTE; aber alles kann er mit seiner HERDE (!) nicht machen – geschweige denn, wenn keine Herde da ist: Dies ist der Fall im Umfeld der Wissenstechniken.

3. Welches besondere Problem lösen Wissens-Techniken? Wie lassen sie sich beschreiben und womit kann man sie vergleichen?
Wissenstechniken lösen das Problem, dass man sich weder auf bestehendes Wissen, noch auf andere Gruppenmitglieder oder irgendeine andere Unterstützung stützen kann. Man steht also quasi alleine da und muss sich selber einen Weg suchen. Wissenstechniken entsprechen also einer universellen Maschine, da sie auf einem trockenen Boden des Wissens genügsam neues Wissen generieren.
Schön. Mit ‚anderer Unterstützung‘ meintest Du sicherlich eine FREMDE (~ nicht sozial bedingte) INFRASTRUKTUR, in die man sich ‚hacken‘ könnte (sorry, ich weiß, böses Wort für Wissenschaft – aber irgendwie gefällt mir der Vergleich ;-)

Programming Literacy:
6. Vergleichen Sie die Ausbreitung der Schriftkunde zur Renaissance mit der gegenwärtigen Lage – sehen Sie Ähnlichkeiten?

In der Renaissance konnte nur eine Minderheit der Menschen lesen und es wurde zunächst auch als unnötig empfunden. („Dafür hat man doch Vorleser“). Clevere Kaufleute machten sich die Schrift zu eigen und hatten dadurch Vorteile.
Heutzutage wird Programmieren als niedere Tätigkeit angesehen, aber vielleicht ändert sich dies auch, da insbesondere Script-Sprachen eine erhebliche Arbeitserleichterung darstellen können.
Super. :-)

Wissens-Techniken vereinfachende Verfahren:

8. Nennen Sie zwei Umstände, welche die Selbstprogrammierung erleichtern, und erklären Sie sie kurz.
Die Frage ist mir nicht ganz klar?
Siehe Folien 25 und 26:
Hier hätte stehen können, inwieweit die Selbstprogrammierung

- durch die Zulässigkeit von (in der Massenfertigung undenkbaren) unperfekten Ergebnissen
- durch die Verfügbarkeit eines Kontinuums von Open Source-Bausteinen

erleichtert wird.

Besondere Werkzeuge:
9. Beschreiben Sie grob, wofür und wie ***Skriptsprachen*** *einsetzbar sind. Beschreiben Sie kurz das Funktionsprinzip.*
Skriptsprachen sind gut für Probleme einsetzbar für immer wiederkehrende Arbeiten oder einfache Aufgaben bei einem großen Aufgabenumfang. Man kann meist mit sehr wenigen Befehlen das gewünschte Ergebnis erreichen und dies sehr viel schneller als die Aufgabe manuell zu bearbeiten.
Beispiel: Das erstellen eines Inhaltsverzeichnis von einem langenText.
Schön. Aber:
Skriptsprachen können sehr wohl MEHR ALS NUR ROUTINEARBEITEN erledigen – sie können alles, was ‚Hochsprachen' können – vielleicht sogar etwas mehr, da sie oft sehr ‚smart' sind. Z.B. hat XOTcl ein fortgeschritteneres Objektmodell als Java. Außerdem sind Skriptsprachen meist TOLERANTER und FLEXIBLER als Hochsprachen.

Der EIGENTLICHE Unterschied ist, dass Skriptsprachen WENIGER FÜR MASSENVERARBEITUNG GEEIGNET sind.

10. Beschreiben Sie grob, wofür und wie ***Programmiersprachen der 4. & 5. Generation*** *einsetzbar sind. Beschreiben Sie kurz das Funktionsprinzip.*
Bei Programmiersprachen der 4. und 5. Generation sagt der Programmierer nur noch was zu geschehen hat und der Computer entscheidet über das WIE. Die Programme sind also als Aufgabenbeschreibungen zu verstehen.
Einsetzbar sind Programmiersprachen der 4. Generation für Sonderbedarfe, wie z.B. die Abfrage einer relationalen Datenbank über SQL.
Schön. Ja, die Sprachen der 4. Generation sind eher SPEZIAL-SPRACHEN.

(Anm: Funktionale Sprachen wurden zur Zeit der ersten Übungen noch nicht betrachtet – sie weisen sowohl Verwandtschaften mit Skriptsprachen wie Programmiersprachen der 4./5. Generation auf.)

11. Beschreiben Sie grob, wofür und wie ***Bayes'sche Netze*** *einsetzbar sind. Beschreiben Sie kurz das Funktionsprinzip.*
Bayes´sche Netze sind stark im Umgang mit unvollständiger und unstrukturierter Kunde, z.B. bei der Analyse von Telefongesprächen.
„Bayes'sche Netze dienen der Repräsentation von unsicherem Wissen und daraus möglichen Schlussfolgerungen. Sie stellen eine spezielle Form der Formulierung von wahrscheinlichkeitstheoretischen Modellen dar.
Ein Bayes'sches Netz ist ein gerichteter azyklischer Graph (DAG), in dem die Knoten Zufallsvariablen und die Kanten bedingte Abhängigkeiten zwischen den Variablen beschreiben. Jedem Knoten des Netzes ist eine bedingte Wahrscheinlichkeitsverteilung der durch ihn repräsentierten Zufallsvariable gegeben die Zufallsvariablen an den Elternknoten zugeordnet. Diese Verteilung kann beliebig sein, jedoch wird häufig mit diskreten oder Normalverteilungen gearbeitet. Eltern eines Knotens v sind diejenigen Knoten, von denen eine Kante zu v führt.
Ein Bayes'sches Netz dient dazu, die gemeinsame Wahrscheinlichkeitsverteilung aller beteiligten Variablen unter Ausnutzung bekannter bedingter Unabhängigkeiten möglichst kompakt zu repräsentieren.
Sind X_1, ..., X_n einige der im Graphen vorkommenden Zufallsvariablen, so berechnet sich deren gemeinsame Verteilung als

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | parents(X_i))$$

Hat ein Knoten keine Eltern, so handelt es sich bei der assoziierten Wahrscheinlichkeitsverteilung um eine unbedingte Verteilung."

Quelle: Wikipedia

Schön. Aber trotzdem nochmal eine Auflistung:
WOZU: Erlaubt Auswertungen, die auf SCHÄTZUNGEN beruhen, was sehr häufig vorkommt (‚klassische' Logikprogrammierung scheitert hier schnell...), z.B. Auswertung geschriebener und gesprochener Sprache
WIE: Ähnlich wie bei Sprachen 4./5. Generation nur Angabe nötig, WAS zu ermitteln ist. Allerdings gibt es in der Regel JEWEILS MEHRERE M÷GLICHKEITEN mit einer bestimmten WAHRSCHEINLICHKEIT – daher erfolgt die Modellierung in Gestalt von Netzstrukturen (siehe Folie 29: „Bayes'sches Netz: Einbruchsmeldung"). Zusätzlich ist noch Parameter- und StrukturLERNEN möglich.
ERLÄUTERUNG:
Erhöhter Aufwand im Vergleich mit Sprachen 4./5. Generation, da i.d.R. keine Eventualitäten ausgesondert werden können (z.B. mit einer Wahrscheinlichkeit von 0,00001%...) – dafür gute Anwendbarkeit von NÄHERUNGSVERFAHREN

12. Beschreiben Sie grob, wofür und wie ***Neuronale Netze*** *einsetzbar sind. Beschreiben Sie kurz das Funktionsprinzip.*
Neuronale Netze sind angelehnt an biologische Gehirne und sind daher einsetzbar in problematischen Bereichen, z.B. bei der Programmierung von Robotern oder von Mustererkennung.
Schön. Aber trotzdem nochmal eine Auflistung:
WOZU: Ausgesprochen VIELSEITIG, insbesonder einsetzbar in vielen Bereichen, wo herkömmliche Programme versagen, d.h., wo kein einfacher klarer Algorithmus gebildet werden kann: Robotik, hohe Komplexität, Unschärfe
WIE: Einerseits wird ein zweckdienliches NEURONALES NETZ modelliert und implementiert (Baukastensystem), andererseits kann es danach (durch die Entwickler oder den LAUFENDEN BETRIEB selbst) noch zusätzliche FERTIGKEITEN durch TRAINING gewinnen.
ERLÄUTERUNG:
Basiert auf Hunderte Millionen Jahre lang bewährtem biologischen Vorbild