A Greedy Knowledge Acquisition Method for the Rapid Prototyping of Knowledge Structures

Claus Möbus
Learning Environments and Knowledge Based Systems
University of Oldenburg
D-26111 Oldenburg, Germany
claus.moebus@uni-oldenburg.de

Heiko Seebold
OFFIS
Escherweg 2
D-26121 Oldenburg
Heiko.Seebold@offis.de

Hilke Garbe
Learning Environments and Knowledge Based Systems
University of Oldenburg
D-26111 Oldenburg, Germany
Hilke.Garbe@informatik.uni-oldenburg.de

ABSTRACT
The main goal of this paper is the presentation of a new
GREedy knowledge Acquisition Procedure (GRAP) for
rapid prototyping of knowledge structures (KS) or spaces.
The classical knowledge acquisition method for this [2] is
even for domain experts cognitive demanding and computa-
tional complex. GRAP interactively generates an online
knowledge acquisition schedule so that experts only have
to provide simple nonredundant judgements about the
(learning / cognitive) precedence in pairs of (learning /
cognitive) objects. From these data GRAP generates a
Hasse diagram of the surmise relation from which the
knowledge structures and optimal user-adaptive learning
paths can be derived. In a case-study we developed with
three expert software engineers a knowledge structure and
optimal learning paths for 23 software design patterns
within a few hours.

Categories and Subject Descriptors
I.2.4 Knowledge Representation Formalisms and Methods
– representation languages, semantic networks, I.2.6
Learning – Knowledge Acquisition

General Terms

Keywords
Knowledge Acquisition; Interactive Greedy Acquisition of
Precedence Relations and Knowledge Structures; Interac-
tive Greedy Construction of Transitive Closures, Hasse
Diagrams and Concept Lattices;

INTRODUCTION
Knowledge Spaces, Concept Lattices and Bayesian Belief
Networks (BBN) are relevant for the success of intelligent
systems in e.g. diagnostics, therapy planning, question
answering and eLearning [1][2][3].

There exist only a few recommendations concerning the
construction of Knowledge Spaces [2, ch.12]. Because of
its cognitive demanding instructions and its runtime com-
plexity these are unsuitable for interactively assessing KS
from domain experts.

This led to the development of GRAP. According to an
interactively generated schedule controlled by GRAP ex-
erts only have to provide simple nonredundant judgements
about the (learning / cognitive) precedence in pairs of
(learning / cognitive) objects. By generating transitive
closures greedily the algorithm controls the selection of
nonredundant pairs, guarantees that the data comprise a
partial order (surmise relation according to [2]) and gen-

erates the Hasse diagram of the surmise relation or the
lattice of the concepts. From these structures optimal user-
adaptive learning path can be derived. In the best case
GRAP acquires the Hasse diagram in just one pass. In this
case the savings in judgements are (1-2/n)*100%, the
judgement complexity is O(n) and the computational
complexity is O(n^3). In the worst case GRAP needs n(n-
1)/2 comparisons. The judgement complexity is O(n^3) and
the computational complexity stays O(n^3).

KNOWLEDGE STRUCTURES
A KS is a pair (Q, K) of solved problems, known items, or
concepts Q and a family K of subsets of Q. The subsets of
K are the knowledge states in the KS. The formal defin-
tion of a KS can be found in [2]. Under the classical ap-
proach the Hasse diagram or the concept lattice has to be
derived by first determining K and then the surmise rela-
tion by using the equivalence q_i ≤ q_j ⇔ K_i ⊆ K_j [2, p.
36], which can be read as: i precedes j iff the set of
knowledge states containing i is a superset of the accord-
ing set containing j. For the above mentioned reasons it is
problematic to derive the precedence judgements from the
set K which has to be acquired directly from experts. In-
stead we leave out the acquisition of K and obtain the
precedence judgements q_i ≤ q_j under the control of
GRAP.

GRAP - A NEW GREEDY METHOD
Its greediness stems from the fact that after each new data
input or after each new inference all possible inferences
are processed. So at any state of the knowledge acquisi-

Copyright is held by the author/owner(s).
K-CAP’05, October 2–5, 2005, Banff, Alberta, Canada.
ACM 1-59593-163-5/05/0010.
tion process only informative new pairs are compared. After the presentation of a pair (i, j) by GRAP subjects have to select a judgement from a set of alternatives (“i causes/precedes j”, “i follows j”, “i neither causes/precedes nor follows j”) internally coded as {+(i, j), -(i, j), (i, j)}. The algorithm works parallel to the main diagonal and if it is possible to sort the items according to some vague ancestral ordering, maximizes the number of possible inferences.

**Tab 1 - GRAP controlled acquisition steps / inferences**

<table>
<thead>
<tr>
<th>No</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>/</td>
<td>+1</td>
<td>+6</td>
<td>++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>/</td>
<td>02</td>
<td>+7</td>
<td>+10</td>
<td>++</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>0</td>
<td>/</td>
<td>03</td>
<td>+6</td>
<td>++</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
<td>/</td>
<td>04</td>
<td>09</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>0</td>
<td>/</td>
<td>+5</td>
</tr>
<tr>
<td>6</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>0</td>
<td>-</td>
<td>/</td>
</tr>
</tbody>
</table>

We demonstrate the algorithm with an example. First we take the KS = {∅, {1}, {1, 2}, {1, 3}, {1, 2, 3}, {1, 2, 4}, {1, 2, 3, 4}, {1, 2, 3, 5}, {1, 2, 3, 4, 5}, {1, 2, 3, 5, 6}, {1, 2, 3, 4, 5, 6}} as the “mental model” of the experts. Nodes are already numbered according a vague ancestral ordering. It is assumed that the experts generate judgements by comparing the set inclusion of the knowledge states according to the equivalence q_i ≤ q_j ⇒ K_i ⊆ K_j. The results are shown in Table 1. Cells marked as <entry> <step> are coded judgements in that order. The content of all other cells is inferred by GRAP’s 13 inference rules (Table 2) which are triggered after any new data entrance in a cell d(i,j), and which can trigger each other recursively. Table 1 shows that we only need 10 judgements; the remaining 5 can be inferred by GRAP. This is a reduction of 33%. Taking only the +(i, j) order information from the transitive closure (Table 1) we are able to reconstruct the Hasse diagram (Figure 1).

**Table 2 – inference rules for controlling GRAP**

<table>
<thead>
<tr>
<th>No</th>
<th>rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+(i, j) ∧ ¬(j, i) ⇒ ¬(j, i)</td>
</tr>
<tr>
<td>2</td>
<td>++(i, j) ∧ ¬(j, i) ⇒ ¬(j, i)</td>
</tr>
<tr>
<td>3</td>
<td>¬(i, j) ∧ + (j, i) ⇒ + (j, i)</td>
</tr>
<tr>
<td>4</td>
<td>¬(i, j) ∧ + (j, i) ⇒ + (j, i)</td>
</tr>
<tr>
<td>5</td>
<td>0(i, j) ∧ ¬0(i, j) ⇒ 0(i, j)</td>
</tr>
</tbody>
</table>

**A CASE STUDY: SOFTWARE PATTERNS**

We used GRAP to find out optimized learning sequences in the domain of software design patterns. In the knowledge acquisition phase GRAP presented pairs of n=23 design patterns [4]. Experts were instructed to state whether either pattern A was a learning prerequisite for pattern B (or vice versa) or whether there was no ordering within this pair. GRAP significantly reduced the maximal number of judgements from 253 (= n(n-1)/2) to 136 (expert C), to 104 (expert B) and to 73 (expert A). So we had savings of 47% - 71%.

**REFERENCES**


