Albert, D., & Kaluscha, R. (1997). Adapting Knowledge Structures in Dynamic Domains. In C. Herzog (Ed.), Beiträge zum Achten Arbeitstreffen der GI–Fachgruppe 1.1.5/7.0.1 ``Intelligente Lehr–/Lernsysteme'', September 1997, Duisburg, Germany [Contributions of the 8th Workshop of the GI SIG ``Intelligent Tutoring Systems''] (pp. 89–100). TU München.

Adapting Knowledge Structures In Dynamic Domains

DIETRICH ALBERT, RAINER KALUSCHA [1]

Institut für Psychologie, Abteilung für Allgemeine Psychologie
Karl–Franzens–Universität Graz, Universitätsplatz 2/III, A–8010 Graz, Austria

E-Mail: dietrich.albert@kfunigraz.ac.at or rainer.kaluscha@kfunigraz.ac.at

Abstract:

For effective processes of knowledge assessment, teaching, information presentation and information use, the information should be presented on basis of a structure which is defined by prerequisite relationships *(Which information do I have to know before I'm able to understand and integrate the newly presented information ?)*. This kind of prerequisite relation for information units has to be determined and tested before using it.

A non-numerical theory for representing these structures is already worked out, and several techniques have been suggested to obtain these structures, e.g.

- mass data collection,
- analysis of didactics and curricula,
- querying experts on prerequisite relationships, competence and performance
- analysis of demands
- analysis of competence and performance,

and also techniques are available to test the obtained structures empirically, which – in principle – is a necessary precondition for using them.

However, domains and thus the related knowledge nowadays change very fast – an impressive example is the internet.

Therefore techniques for convergent structuring, for dynamic testing and for dynamic refinement have to be developed. This article will give an overview of existing techniques in the static case, i.e. the domain doesn't change at all or only very slowly, and discuss advancements towards the use of these techniques in dynamic domains.

[1] alphabetical order, send requests for reprints to either author

1. Introduction

Due to modern communication techniques, a vast amount of information and knowledge has become accessible for many people. But to find the relevant information in the floods, is still a difficult task. Search engines are not of much help, as they look only for keywords, and don't take into account the inherent structure of the information and the knowledge.

Only dividing the knowledge in small parts and structuring it leads to an easy to understand way of presentation. We'll call these units of information (sometimes also referred to as learning objects) *items*. For example, these items may be facts, concepts, problems or skills from a given domain of knowledge.

After defining a domain of knowledge and identifying the relevant items within the chosen domain, the question arises whether these items may be presented independently from each other or not. Usually, relationships exist between items, e.g. *prerequisite relationships* or *surmise relationships*. Roughly speaking, a prerequisite relationship means that people – as a necessary precondition – do have to know about one or more items in order to master another item. A surmise relationship indicates that a person who understands about a certain item will (probably) know about one or more other items.

The theory of knowledge spaces (Doignon & Falmagne, 1985) provides a formal framework (based on the theory of orders and lattices) for the representation of such relationships.

2. Knowledge Structures

Several ways for representing knowledge structures in mathematical models are known, e.g. *knowledge spaces* and *surmise systems* (Doignon & Falmagne, 1985), *and–or–graphs* (Nilsson, 1980) or *implication relations* (Dowling, 1993). All of these representations have shown to be equivalent because bijective mappings between them exist.

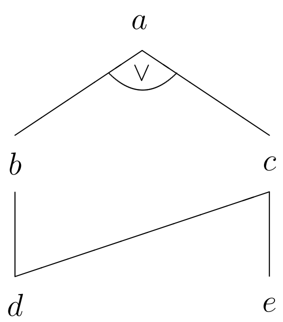

Figure 1: Example of an *and–or–graph*

Figure 1 shows an example of an *and–or–graph* of a domain $Q := \{a, b, c, d, e\}$ with five items. *Or–nodes* are marked by a curved line and a $\vee$ symbol, e.g. to master item a it is sufficient to know about item b *or* item c (or both of them). *And–nodes* don't have a special mark, e.g. a person has to understand about both items d *and* e in order to master item c. In this sense, items d and e are *prerequisites* for c. Vice versa, knowing a person masters item c, it can be *surmised* that this person will also master items d and e.

The corresponding *knowledge space* $\mathcal{K}$ is the set of all subsets K of the domain set Q which conform to the relationships mentioned before:

$$K := \{\emptyset, \{d\}, \{e\}, \{b,d\}, \{d,e\}, \{a,b,d\}, \{b,d,e\}, \{c,d,e\}, \{a,b,d,e\}, \{a,b,d,e\}, \{b,c,d,e\}, \{a,b,c,d,e\}\}$$

For example, the subset $\{b\}$ is not a member of the knowledge space, since item d is a prerequisite for item b, which yields $\{b,d\}$ as the smallest (with regard to the ordering given by set inclusion) member of the knowledge space $\mathcal{K}$ which contains item b. The members of the knowledge space are called *knowledge states*.

A knowledge space $\mathcal{K}$ on a set of items Q has (by definition) the following three properties:

$$P1 : \emptyset \in \mathcal{K} \qquad P2 : Q \in \mathcal{K} \qquad P3 : \bigwedge_{K_1, K_2 \in \mathcal{K}} K_1 \cup K_2 \in \mathcal{K}$$

$P1$ and $P2$ make sure that the perfect novice, who doesn't master any item, and the perfect expert, who knows about all the items in the domain Q, are represented in the knowledge space. $P3$ means a knowledge space is closed under union. An interpretation of this could be that two persons in the knowledge states K_1 and K_2 may share their knowledge and teach each other, so that both end up in the state $K_1 \cup K_2$ which is again a member of the knowledge space.

Please note that these knowledge structures do not only give insight which paths a person can follow when moving within a domain, they also allow for the adaptive and efficient assessment of knowledge (Falmagne 1985, Dowling & Kaluscha, 1995).

Knowing that these knowledge structures are useful for knowledge assessment, teaching and information presentation, techniques are needed for establishing them. Several approaches are available and will be discussed in the following sections.

Up to now only the static case has been discussed, i.e. the domain has been fixed, and the structure of a fixed set of items had to be determined. In this paper, also the case of more dynamic domains, where

1. items [2] may become obsolete or
2. new items enter the domain from time to time

will be discussed.

3. Mass Data Collection

One approach is to collect answer patterns in a given domain from a large number of subjects. But due to the huge number of possible knowledge states which may increase

[2] We will not discuss the trivial case where the obsolete item or a new item is equivalent to other items in the domain. New or obsolete items should be considered as new or obsolete item types or item classes.

exponentially with the number of items considered, this method is rarely feasible. Furthermore, the resulting knowledge structures will be extremly dependent on the sample, and thus might be no longer valid when applied to other populations.

However, if the data can be collected without too much effort, and the domain does not consist of a large number of items, analyzing mass data might be worth trying.

For example, in the state of New York passing the Regents Competency Test in Mathematics is a requirement for graduation from high school.

Villano (1991) chose 20 items from this test and constructed a knowledge structure both by analyzing the answer patterns of 67,204 participants in the test and the judgements of experts.

However, in a dynamic domain, mass data collection has an important drawback. If an item has to be deleted, there's no problem – the answer patterns without this item may be used for a new analysis. But if a new item appears, retesting the original sample with the new item would be inevitable which often is not feasible. In the example of the New York Regents Competency Test this enterprise would mean to contact all 67,204 students that took part in the test in 1987, collect their answers to the new item and insert the result in the individuals' answer pattern. Furthermore, these answer patterns are often collected anonymously for the protection of the participants, so that it wouldn't be known which new result belongs to which answer pattern.

4. Analysis Of Didactics and Curricula

This approach hasn't been worked out scientifically yet. To our knowledge, researchers in didactics haven't yet formulated a formal model for their results suitable for deriving relationships between items.

The idea is to use knowledge from didactics and analyze existing curricula for a given domain.

Once the order in which items from the domain are taught is known, inferences concerning prerequisite relationships can be drawn.

For example, if item a is presented after item b, obviously a cannot be a prerequisite for b. On the other hand, an item taught shortly before another is liable to be (one of) its prerequisite(s).

In the dynamic case, new items have to be integrated in existing curricula. From the position where they're placed probable prerequisite relationships can be identified, as discussed above. For updating the knowledge structure the same techniques could be used as described in the next section.

5. Querying Experts

One way of determining a knowledge structure for a given domain is to query experts on prerequisite relationships (Falmagne et al. , 1990; Dowling 1991, Koppen 1993, Kaluscha 1994).

In the first step, the domain of knowledge to be investigated has to be defined and the relevant items within the domain have to be identified, usually with the help of experts in the field.

In a second step, the relationships between these items have to be determined. Please note that the expertise needed for the first and the second step may be different – defining the domain is a task for experts from the field, while determining the relationships between items usually requires additional psychological and pedagogical knowledge as well as experience in teaching the items (for an example see Held, Schrepp and Fries, 1995). For instance, if the domain chosen was elementary mathematics, mathematicians could be a help for accomplishing the first step, while experienced mathematics teachers should be preferred as experts in the second step.

A computer based querying procedure has been implemented by Kaluscha (1994) based on the algorithm due to Dowling (1993). Koch and Quante have improved the procedure and provided an enriched user interface for the experts (Dowling, Koch and Quante, 1996).

As the experts usually are not able to give the complete list of prerequisite relationships, which is due to the huge number of possible relationships even in small domains, the experts have to judge *standard form assertions*, e.g.

> `Imagine a person who does not master the items` $p_1, \ldots, p_k$.
> `Is it then (practically) certain that this person does not master item` q ?

The items $p_1, \ldots, p_k$ are called the *premise (set)*, and the item q is called *consequence*. The index k is called the *size of the premise*. For judged assertions, the index k takes values between 1 and $N-1$ where $N := |Q|$ is the total number of items considered.

The experts either accepts or rejects these assertions. By the rules of mathematical logic, inferences can be drawn from previous judgements, and thus the number of standard form assertions to be presented to the experts can be considerably reduced (Falmagne et al., 1990; Koppen, 1993; Dowling, 1993). If a suitable querying strategy is used which is optimized to present assertions for judgement which allow for a large number of inferences the necessary number of judgements is again considerably reduced (Dowling & Kaluscha, 1995). The querying procedure finishes when all standard form assertions have been judged by the expert or by inference from previous answers of the expert.

The first dynamic case, to remove an obsolete item i_{obs} from a knowledge structure, is a rather simple task if a knowledge space $\mathcal{K}$ is chosen as representation of the knowledge structure:

> Remove from all knowledge states $K \in \mathcal{K}$ the obsolete item i_{obs} (and eliminate duplicate knowledge states). [3]

This gives immediately a new knowledge space $\hat{\mathcal{K}}$, from which the other representations can be derived (Dowling, 1993; Koppen, 1993). Please note that the algorithm mentioned above preserves the three properties $P1$, $P2$ and $P3$ of a knowledge space, i.e. $\hat{\mathcal{K}}$ is indeed a knowledge space on the domain $Q \setminus \{i_{obs}\}$.

In the second dynamic case, adding a new item i_{new} is a more complex problem. In order to preserve property $P3$ (closedness under union) the introduction of the new item can result in two times as much knowledge states as before.

[3]This is, by the way, the trace of $\mathcal{K}$ on $Q \setminus i_{obs}$ (Falmagne & Doignon, to appear: Def. 7)

On the one hand, if we choose a simple approach and assume there are no relationships between the new item i_{new} and the other items $i_k \in Q$, for each knowledge state $K \in \mathcal{K}$ a knowledge state $\hat{K} := K \cup \{i_{new}\}$ has to be added to obtain the new knowledge space $\hat{\mathcal{K}}$. So we end up with twice as much knowledge states.

On the other hand, it wouldn't be feasible to start the querying procedure all over again, as the experts are rare and their time is too valuable.

So we choose another approach. The querying procedure virtually starts from scratch, but only those questions, i.e. standard form assertions, have to be presented to the expert for judgement which contain the new item i_{new} in the premise or consequence – the answers to the other questions are already known from the previous run of the querying procedure on the domain Q. Furthermore, not all of the standard form assertions containing the new item i_{new} have to be judged by the expert, as judgements can be inferred from the expert's answers to previous questions containing i_{new}.

Thus, already a partial run of the querying procedure yields the new knowledge space $\hat{\mathcal{K}}$ on the domain $Q \cup \{i_{new}\}$.

If several items have become obsolete, or several items have to be newly added, the procedures above can be performed for each obsolete or new item respectively in sequence.

Another possibility for adding several new items at one time could be to first determine the knowledge structure within these new items, and then use techniques for merging two or more existing knowledge structures into a new one which are under development (Falmagne & Doignon, to appear).

6. Analysis Of Demands

Another approach is the analysis of demands that an item imposes on a person (Albert & Held, 1994; Held, 1993, Lukas & Albert, 1993) [4] . For instance, demands can be applying a skill or knowing some facts. The demands can be obtained by e.g. content analysis, didactical knowledge or analysis of the underlying cognitive processes (Schrepp, 1993).

For example, if the domain chosen was computing expressions of the form $a + b \times c$ with $a, b, c \in \mathbb{N}_0$, an example for an item would be the task "compute $1 + 2 \times 3$".

Relevant demands of this kind of tasks are

1. A: adding natural numbers
2. M: multiplying natural numbers
3. P: knowing that multiplication takes precedence over addition if no parentheses are specified

These demands are called *(problem) components*. A component may have several *attributes* (values) from a *component set*. These attributes can be interpreted as a kind of difficulty level.

If component A (adding natural numbers) is considered the following cases may occur:

[4] These papers deal with a subclass of knowledge structures called the quasiordinal spaces which correspond to surmise relations.

1. a_0: $a = 0$, i.e. no addition is required
2. a_1: $a + b \times c < 10$, i.e. only one digit numbers have to be added without considering a carry
3. a_2: $a + b \times c \geq 10$, i.e. only one digit numbers have to be added with considering a carry
4. a_3: two digit numbers are involved
5. a_4: three and more digit numbers are involved

Similiar considerations can be made for component M (multiplying natural numbers) and for component P (knowledge on precedence rule), where the precedence rule may have to be applied or not. Thus each problem can be characterized by a vector of components and their attributes, e.g. the problem "compute $1 + 2 \times 3$" has the following attributes:

- a_1: $1 + 6$ requires no carry operation
- m_1: 2×3 is a multiplication of one digit numbers, resulting in a one digit number
- p_1: the precedence rule has to be applied

Thus the problem is characterized by the vector (a_1, m_1, p_1) which is a member of the Cartesian product of the component sets $A \times M \times P$.

If a partial order (with respect to "difficulty" or prerequisite relations between attributes) exists on each of the component sets, a partial order on the items can easily be derived. An item a is considered a prerequisite for another item b if and only if the attribute value of each of its components is a prerequisite for the corresponding attribute value of the components of the other item (*coordinatewise order*, see Davey and Priestley, 1990, p. 18). For example, let the three partial orders on the components addition, multiplication and precedence be given by

$$a_0 \preceq a_1 \preceq a_2 \preceq a_3 \preceq a_4 \qquad m_0 \preceq m_1 \preceq m_2 \preceq m_3 \preceq m_4 \qquad p_0 \preceq p_1$$

Then a problem of type $\{a_2, m_1, p_1\}$ will be a prerequisite for a problem of type $\{a_3, m_3, p_1\}$ because the prerequisite relationship holds for each component. However, two problems of the types $\{a_0, m_2, p_0\}$ and $\{a_1, m_1, p_1\}$ have no relationship at all, they cannot be compared.

Alternatively, if the components are not of equal importance, a *lexicographical order* can be used (see Davey and Priestley, 1990, p. 19).

For other domains, the ordering principles of *set inclusion* or *sequence inclusion* may be more appropriate (see Albert & Held, 1994 for examples).

The order on the items gives the relationships between items, which can be used the same way as those obtained by querying experts.

What might happen in the dynamic case ? If an item disappears, the task is simple. But if a new item i_{new} has to be added several cases have to be considered:

1. The new item can be characterized by the existing components and attributes. No special action has to be taken, the new item is simply integrated into the ordering of the other items according to its characterizing vector of components and attributes and the ordering principles.

2. The new item induces new attributes on some components, i.e. the component sets grow for some components. For these components, the partial order has to be redefined. While the basic structure of the order of attributes may be preserved, just the new attribute values have to be inserted in the correct position. After that, while the order on the old items stays the same, the relationships between the new and the old items can be derived by the ordering principles mentioned above.

3. The new item i_{new} requires adding a new component C to the structure. Even in this case there are no real difficulties. First all old items have to be analyzed what attribute values they show with respect to the new component. Then two possible cases have to be discussed:

   - For all of the old items, the attribute value of the new component C is the same, e.g. the component isn't present. Then the component set is defined to be $\{c_0, c_1\}$ where c_0 means the component is not present, and c_1 is the attribute value of the new item with respect to the new component. Then the partial order on C is given by $c_0 \preceq c_1$, and again, the order on the items can be derived by the algorithms mentioned above.
   - If the new item does not only induce a new component C, but also gives rise to the insight that the new component has been neglected on the other items, i.e. the old items have different attributes c_i on the component C an appropriate partial order has to be defined on the component set C, and then the relations between the items can be generated by the ordering principles mentioned above.

7. Competence and Performance

Korossy (1993, 1997) extended the theory of knowledge structures by separating *competence* and *performance*. Competence means skills or abilities that enable a persons to solve a problem, and cannot be observed directly. Performance is the behaviour, e.g. the answer that is given, and can be observed.

Of course, competence, demands and performance are related. However, competences are properties of persons, while demands are properties of problems. While a demand requires a competence to fulfill it, the relationship is not a one to one relation. For example, the demand "add two natural numbers" may be met by adding mentally, by adding with the help of paper and pencil or by operating a pocket calculator.

Similarly, when a performance of a person is observed, it is not obvious what underlying competences have contributed to the solution.

Thus Korossy introduced two spaces: a competence space on a set of (elementary) competences, and a performance space on a set of items. These spaces have the same properties as knowledge spaces, and also prerequisite and surmise relations exist. A state in the competence space describes the competences a person has, while a state in the performance space is given by the set of items that the person can master.

By identifying the relationships between competences and performance on the items, an *interpretation function* and a *representation function* can be defined. The interpretation function maps a problem or an item to the set of all competence states that allow for solving the problem. Vice versa, the representation function maps a competence state to the set of problems which can be mastered with the competences of this state (Korossy 1993, 1997). Please note that it is sufficient to determine one of these functions as the other one can be derived (Korossy, 1996).

In any case, the interpretation function (or the representation function) has to be adjusted when items become obsolete or are inserted into the performance space. The same holds for new competences that might be added to the set of competences, enlarging the competence space. This is simply due to the fact that the domain of the functions changes.

If a new competence is added, the procedure and the problems for determining the new competence space are similiar to that for knowledge spaces as relationships between the new and the old competences should be used to avoid a large increase of the number of competence states. Furthermore, for each of the items has to be checked whether the new competence opens up new ways of solving the item. If so, the interpretation function (or the representation function) has to be updated because now there are more competence states which allow for solving this item. For example, in the elementary school pupils are forbidden to use pocket calculators. Later in high school the competence "using a pocket calculator" comes into play. This competence enables the pupils to solve items of the type "addition of natural numbers", but also "addition of real numbers". This means the interpretation function has to be changed for both item types.

For deleting a competence from the competence space, the same procedure as deleting an item from a knowledge space is used (see section 5).

If a new item is added to the domain, the following cases may occur:

1. The new item can be solved with the existing competences. In this case, the value of representation function has to be augmented by the new item for all competence states that allow for solving the new item. This means that the result of the interpretation function for the new item will be the set of all competence states that allow for solving the new item, as expected.

2. Mastering the new item requires new competences. Now both the steps for adding new competences (see above) and a new item have to be performed.

Schrepp (1993) goes one step further than Korossy and investigates not only the competences of a person, but also the underlying cognitive processes.

A model for these processes also allows to derive the competence structure, and by means of the representation function the performance structure can be obtained (see Schrepp, 1993).

The neccessary actions when a new item arises depend on whether the model of the cognitive processes can explain how a person masters the new item. If so, the new structures can be derived easily. Otherwise, the cognitive model has to be extended which may require further research.

8. Conclusion

Several approaches for determining knowledge structures are known. Though up to now they've been designed for and applied to static domains only they (except for mass data collection) also can be used in domains that are changing dynamically.

The extent to which old results can be reused and the amount of reanalysis needed varies between the different techniques, also depending on the properties of the new item types that occur.

Most of the necessary work can be done by algorithms which have been sketched in this paper. However, they have to be worked out in detail and implemented on computers. Then pilot studies, applying the techniques in dynamic domains, have to be carried out to gather more experience and prove the usefulness of the dynamic approach.

References

Albert, Dietrich and Held, Theo (1994). Establishing knowledge spaces by systematical problem construction. In Dietrich Albert (Ed.), *Knowledge Structures* (pp. 78-112). New York: Springer Verlag .

Albert, Dietrich; Schrepp, Martin and Held, Theo (1994). Construction of knowledge spaces for problem solving in chess. In Fisher, G.H. & Laming, D. (Ed.), *Contributions to Mathematical Psychology, Psychometrics, and Methodology* (pp. 123–135). New York: Springer-Verlag .

Davey, B. and Priestley, H. (1990). *Introduction to lattices and orders.* Cambridge: Cambridge University Press .

Doignon, Jean–Paul, & Falmagne, Jean–Claude (1985). Spaces for the assessment of knowledge. *International Journal of Man-Machine Studies*, **23**, 175–196.

Dowling, Cornelia (1991). *Constructing knowledge structures from the judgements of experts.* Habilitationsschrift, Technische Universität Carolo-Wilhelmina, Braunschweig, Germany.

Dowling, Cornelia (1993). Applying the basis of a knowledge space for controlling the questioning of an expert. *Journal of Mathematical Psychology*, **37**, 21–48 .

Dowling, Cornelia and Kaluscha, Rainer (1995). Prerequisite relationships for the adaptive assessment of knowledge. In Jim Greer (Ed.), *Proceedings of the AI–ED 95*, 43–50 .

Dowling, Cornelia; Koch, Uwe and Quante, Kai A. (1996). A new interface for querying experts on prerequisite relationships. *Proceedings of the OzCHI'96*, Falmagne, Jean–Claude & Doignon, Jean–Paul (to appear). Meshing knowledge structures. In Dowling, Cornelia; Roberts, Fred and Theuns, Peter (Eds.), *Recent Progress in Mathematical Psychology* Hillsdale, USA: Lawrence Earlbaum Associates Ltd. .

Falmagne, Jean–Claude; Koppen, Mathieu; Villano, Michael; Doignon, Jean-Paul and Johannesen, Leila (1990). Introduction to knowledge spaces: how to build, test and search them. *Psychological Review*, **97**, 201–224 .

Held, Theo (1993). *Establishment and empirical validation of problem structures based on domain specific skills and textual properties.* Doctoral Dissertation, Universität Heidelberg, Germany .

Held, Theo; Schrepp, Martin and Fries, Stefan (1995). Methoden zur Bestimmung von Wissensstrukturen — Eine Vergleichsstudie (Methods For Determining Knowledge Structures — A Comparison Study). *Zeitschrift für Experimentelle Psychologie*, **XLII, 2**, 205–236 .

Kaluscha, Rainer (1994). *Ein effizienter Algorithmus zur Expertenbefragung (An efficient algorithm for querying experts).* Diplomarbeit, Technische Universität Carolo-Wilhelmina, Braunschweig, Germany.

Koppen, Mathieu (1993). Extracting human expertise for constructing knowledge spaces: an algorithm. *Journal of Mathematical Psychology*, **37**, 1–20.

Korossy, Klaus (1993). *Modellierung von Wissen als Kompetenz und Performanz (Modeling Knowledge as Competence and Performance).* Doctoral Dissertation, Universität Heidelberg, Germany .

Korossy, Klaus (1996). *A qualitative–structural approach to the modeling of knowledge.* Report of the Institute of Psychology, University of Heidelberg, Germany .

Korossy, Klaus (1997). Extending the theory of knowledge spaces: a competence–performance approach. *Zeitschrift für Psychologie*, **205**, 53–82 .

Lukas, Josef and Albert, Dietrich (1993). Knowledge assessment based on skill assignment and psychological task analysis. In Strube, Gerhard and Wender, Karl (Eds.), *The Cognitive Psychology of Knowledge, volume 101 of Advances in Psychology.* (pp. 139–160). Amsterdam: North–Holland .

Nilsson, Nils J. (1980). *Principles of artificial intelligence.* Palo Alto: Tioga Publishing Co.

Schrepp, Martin (1993). *Über die Beziehung zwischen kognitiven Prozessen und Wissensräumen beim Problemlösen (On the relatinship between cognitive processes and knowledge spaces in problem solving).* Doctoral Dissertation, Universität Heidelberg, Germany .

Villano, Michael (1991). *Computerized knowledge assessment: Building the knowledge structure and calibrating the assessment routine.* Doctoral Dissertation, New York University, New York. *Dissertation Abstracts International*, **552**, 12B.