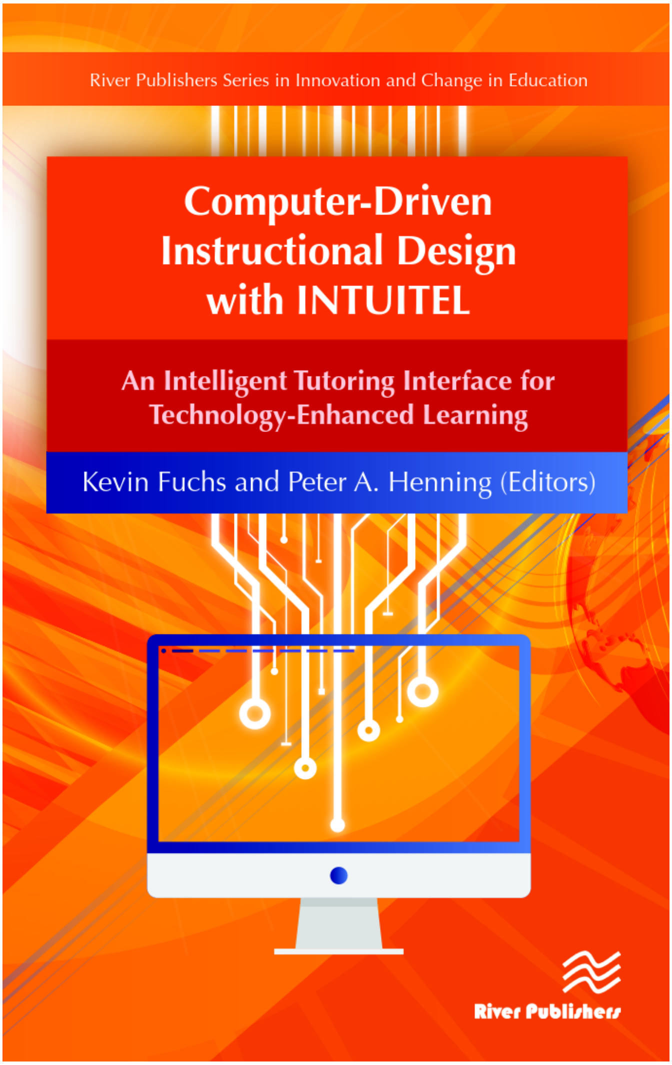

3.3 Learning Progress and Learning Pathways

Alexander Streicher and Florian Heberle

The central information mediating component of INTUITEL is the Learning Progress Model (LPM) module, which connects all other components. It acts as a preliminary stage for the INTUITEL Engine. By providing functions to perform transformations of learner scores, history, and pedagogical as well as domain knowledge into a position within a cognitive space, the LPM prepares the data to be usable by the semantic reasoner in the INTUITEL Engine.

In order to achieve this, it relies on multiple data sources which carry data about the users. First, there is the data about the learner as represented in the LMS. Secondly, the pedagogical and domain knowledge in form of the Pedagogical Ontology and the respective Cognitive Models, provide information about the actual learning content and how to guide a learner through the learning material. And in the third place, a user model of internal INTUITEL relevant user data like session data or the navigation history. The specific tasks of the LPM regarding the recommendation creation process can be summarized with three core themes:

1. Data aggregation
2. Data storage
3. Data transformation

To complete its range of functions, the LPM includes components to trigger direct user feedback before the reasoning process starts, determine the most suitable Learning Pathway (LP) and create the optimized data basis for the Reasoning Engine. Explicitly not task of the LPM is finding suitable Learning Objects (LO) for a learner. This is solely task of the INTUITEL Engine. The LPM is only indirectly involved in this process since it provides and edits the relevant information to be later used by the INTUITEL Engine.

3.3.1 INTUITEL Recommendation Process

There are multiple components that are part of the recommendation creation process and the individual tasks need to be distributed between them. The LPM is one of these components and the following descriptions show which tasks are conducted by the LPM and which tasks the other modules take over. This also depicts how the LPM is positioned in the INTUITEL overall system design.

The functional process has three stages, as depicted in Figure 3.2. There firstly is the stage of data preparation where the LPM collects and reformats the data for the INTUITEL Engine. Secondly, the reasoning process is conducted to determine the elements in the different result sets. Thirdly, the results are sorted and reformatted to be in a format that is compatible with the rest of the INTUITEL system. Also, learning recommendation messages are created to guide the LMS user.

3.3.2 Comparison to Real-Life Tutoring

Speaking in terms of a real-life learning scenario, INTUITEL is a workflow engine to process well established learning methods, in particular by deducing relevant and individual recommendations for learners. Without loss of generality, Table 3.2 outlines the analogy between the decision process of a real teacher and INTUITEL.

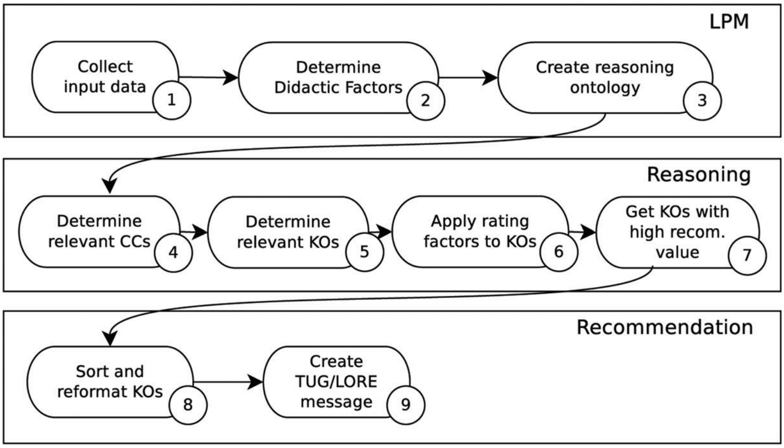

Figure 3.2 Simplified functional procedures for the recommendation creation in the back end.

Table 3.2 Comparison between learner and INTUITEL behavior in learner counseling

Pedagogical Aspect	INTUITEL Analogy
A human teacher senses when a learner needs assistance. The sensing process may rely on subconscious experience based processes as well as involve cognitive reasoning	Via the LMS-interfaces, INTUITEL registers and protocols the current and precious states of the learner. The LPM serves as the sensing instance, either by simple rules based inference or by more complicated ontology based reasoning.
In looking at test results and carrying out a spoken dialog, a teacher tries to assess the learner's current needs.	Usage of personalized information INTUITEL has collected for each learner and request of up-to-date information via USE or TUG dialog.
By applying personal didactic/domain knowledge or by intuitive reaction, the teacher forms an opinion about the learner's situations.	Application of transformation rules to create the different Didactic Factors and their integration in the learner-specific Learner-State-Ontology.
The teacher evaluates the situation on a topical basis and aligns it to his or her personal knowledge about the topic.	With the Cognitive Model stored in the SLOM file, INTUITEL draws conclusions about necessary alignment of the learning process.
The teacher ponders which advice fits best.	Reasoning process in the INTUITEL Engine to select suitable KOs and/or the decision which natural language feedback should be given.
Teacher gives the learner an advice which fits his/her current situation best.	Recommendation of specific content via the LORE interface and/or natural language messages via the TUG interface.

3.3.3 Reflex Reactions

Like a human teacher the INTUITEL system should always be able to react to the learner. At best, the recommendation process is always based on sound considerations of all input information. But sometimes even a real teacher is not able to find the best recommendation to help the learner. Possible reasons for that could be the lack of some specific knowledge or the deduction problem is too complex to be solved in an appropriate time. In such cases teachers must rely on their intuition. Their "reflexes" kick in and the teachers give nearly-as-good answers as recommendations.

Ordinarily, the more experience a teacher has, the better the recommendations fit the learners' situation and needs. INTUITEL aims to do the same in order to provide for the best possible guidance. It thus needs a mechanism which decides when to bypass the reasoning process and when to trigger a direct reaction. Therefore, the LPM contains a "reflex module". Like in the real-life scenario, the reflex module acts when it is foreseeable that a response is not going to be created in time or when it is obvious that a certain action has to take place.

Because a recommendation cannot necessarily be created in all cases, the reflex module triggers a message that either informs the learner that his recommendation is pending (e.g., "I am still thinking about it, just one moment please.") or enforces the creation of a specific question that the learner should answer (e.g., "Which Learning Pathway do you want to choose? Select one of the following"). Another example would be to create a welcome message, if the learner just started a new session. For such a message, no computationally expensive reasoning has to take place.

3.3.4 LPM Input

This section defines and explains the different input types for the LPM. It firstly describes the learner dependent set of data. Further, the concept of Learning Pathways and their realization in the Pedagogical Ontology is outlined. Afterwards, the concept of the Cognitive Models is described.

3.3.5 Learner Input

Learner input basically describes all personalized learner information INTUITEL collects from and via the LMS and which can be used as a source for the recommendation creation process. This not only includes learner scores in terms of grades or results of tests, but also additional data which can be collected by INTUITEL. There is a multitude of different types or kinds of learner scores that provide diverse information about the learner from a learning habit, a learning progress, and a situational perspective. Depending on the possibilities of the LMS and the available information, some of it might also be collected by directly asking the learner. Examples of the available learner data are, apart from the rather obvious question how good the learner is in terms of grades, amongst others:

- What and when did the learner access content?
- In which order did the learner access learning content?
- How long was the learner working on certain a Learning Object?

- Which kind of device is the learner currently using?
- How good is the learners Internet connection?
- Is the learner currently in a noisy environment?
- Other possible raw data items may indicate emotional or stress measurements involving bodily interfaces to the learner.

In short, learner input describes all information on all aspects regarding how and what a learner currently and also previously has learned in the eLearning system across all courses. For a detailed list and description which data can be accessed through the respective INTUITEL interfaces, see the INTUITEL data model in Section 3.5. Examples could be age, name, gender etc.

3.3.6 Pedagogical Input

The pedagogical input for the LPM consists of two parts. On the one hand, there is the terminology specified in the Pedagogical Ontology. It provides a comprehensive set of entities for the modeling of learning material in INTUITEL. The structuring of learning material, independent of its actual LMS realization, allows a semantically rich description of e-learning courses. The central elements, namely CCs for abstract topic-based structuring of courses and KOs for the description of the actual content, allow the LPM to understand the meaning of the Learning Objects in the course. However, what is more important for the pedagogical input is the second aspect, the availability of Learning Pathways. In order to guide learners through an e-learning course, INTUITEL needs a "map" describing how to find a suitable route. This basically is what Learning Pathways (LP) provide for the back end. Modeling Learning Pathways with them, the LPM and the INTUITEL Engine are able to deduce a didactically reasonable route through the learning material.

3.3.7 Domain Input

The previously described pedagogical input alone is as itself insufficient for the back end, since it only provides the technical and didactic foundations. The description of the actual learning content is missing so far. In order to include this, the LPM needs the description of the knowledge domain from the lecturer. This is provided through the Cognitive Model (CM) which is created by a domain specialist. The CM outlines the curriculum of a certain domain of knowledge. This OWL-based description specifies which CCs are available in a specific Knowledge Domain (KD). Based on this a course in a LMS is connected by completing the course specification with the semantically rich description of the KOs. The thereby defined meta-data contains detailed information about the learning material itself (e.g., the contained media or what type of knowledge they represent).

To summarize, the domain input subsumes all information which the LPM and INTUITEL Engine need to understand the internal coherences in an eLearning course.

This does not mean that INTUITEL understands the content itself, i.e., what the learner is trying to learn, but is able to react on the semantic data as mentioned before.

A figurative example would be the simplified e-learning course "Math for absolute beginners" (see Figure 3.3). Consisting of a number of pages introducing to the basics of what numbers are and the basic arithmetic operations addition and subtraction, it depicts a very brief example of domain input in INTUITEL. On the first level, there is the start point of the course itself, the Knowledge Domain (KD). Branching from there, the individual CC are connected via macro LPs and allow navigating between them in a didactically reasonable way. The KOs is attached to the CCs and the micro LPs specify how a learner should best range between them on basis of the KO meta-data. With this information, the back end can deduce an optimal route for each individual learner, based on how many different LPs are available for that specific course.

Although this is not categorized as domain input, it should be noted that the individual Learning Pathways between the elements are firstly possible because the domain input provides a description of the content between which a route is possible. INTUITEL can thereby rely on the different macro LPs that the tutor has added into the Cognitive Model and also the micro LPs that are available due to their definition in the Pedagogical Ontology.

3.3.8 Set-based Rating of Learning Objects

Due to the usage of OWL-reasoning in INTUITEL, the approach of how recommendations are created is different to how other software solution approach that task. One of the main features of OWL-reasoners is the identification of elements in a set. This is a very useful functionality for INTUITEL, because creating a KO recommendation can be interpreted as the task of finding the elements of the set that contains the optimal KOs for the learner. Thus, the fundamental question of the back end is: What are the defining criteria of this set and how to evaluate them? Building on the previous introductions on what input is available for the LPM, it can be registered that there basically are three influential factors concerning the suitability of a KO:

- the learners macro Learning Pathway
- the learners micro Learning Pathway
- the situational dependent information of the learner

The first two points are relatively unproblematic, because this basically only requires data that is already available in the present system. This is at first the data which specifies the Learning Objects that are part of the course. Secondly, the history of the learner is needed to filter those LOs that are not available for the recommendation (e.g., because they are already finished). Thirdly, the learners personal Learning Pathway is required, to find the LOs that are next in the sequence. Given that this information is handed over to the Engine in a suitable way, the LOs that are relevant in context of the learners LP can be identified. This is possible, because the present problem

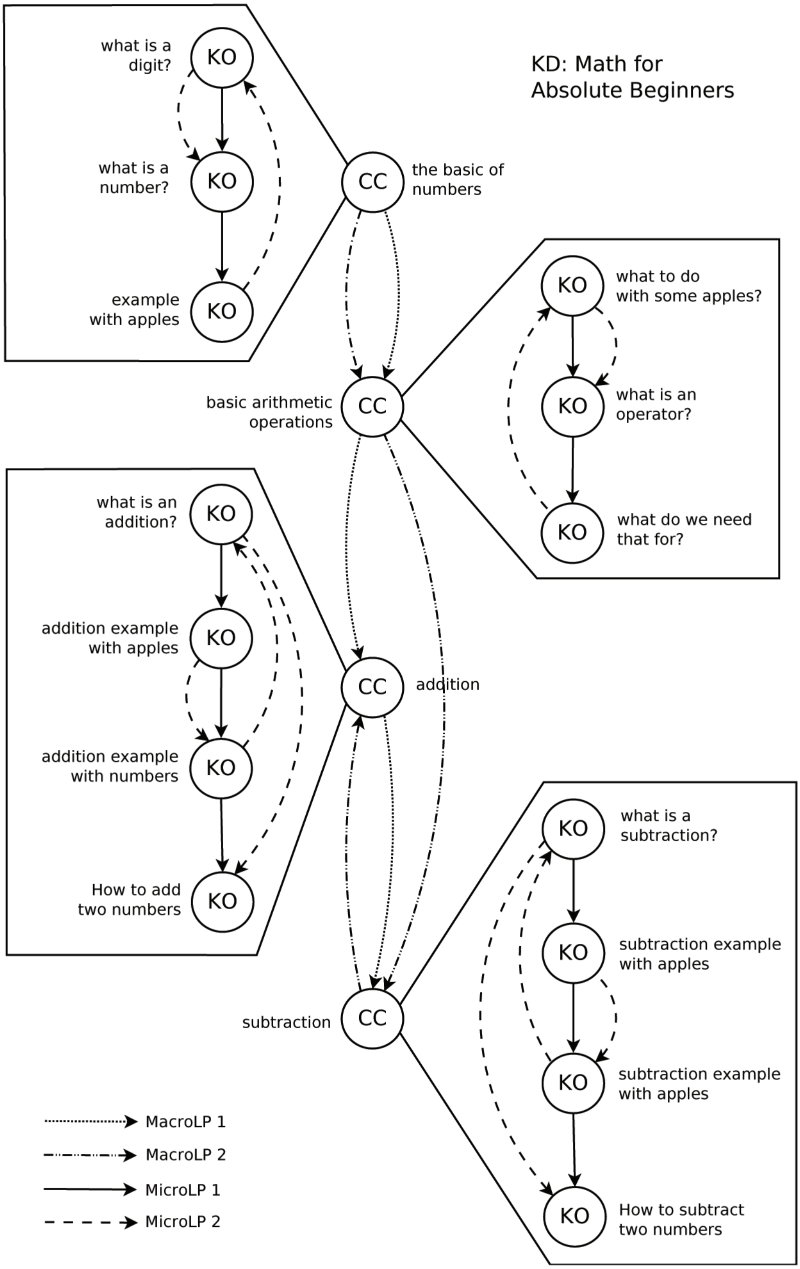

Figure 3.3 Exemplary different LPs diagram showing the coherences in the domain input, including different LPs.

can be described on basis of sets, allowing the Engine to iteratively reduce the number of LOs. In order to understand this procedure, it is advisable to illustrate this step-by-step.

As the first step, it should be clear that a LO is an umbrella term that combines CCs and KOs (for the sake of simplicity, Knowledge Domains (KDs), which are by definition also LOs, are excluded here). Each LO is thus either a CC or a KO. So, when starting with a set that contains all LOs of a certain course, it is possible to segment it into the set of CCs of that course and into the set of KOs of that course. These two sets can further be segmented into sets that differentiate LOs depending on whether they are rated as unseen, unfinished or already completed (Figure 3.4).

Independent of these sets is it possible to subdivide the set of available LOs of a course regarding their distance to the current Learning Pathway positions. For this, firstly the set of currently active KOs is identified which is represented by the small dark circles in Figure 3.4. This set of currently active KOs is either empty or contains exactly one item. It is empty if the learner has never worked on the course and has not yet accessed a KO in the current session. If the learner has already accessed a KO in the LMS, the set contains exactly this KO (i.e., the last one), which is rated as either unfinished or already complete.

Based on that, it is possible to identify the set of current CCs, which contains all CCs that has reference on the currently active KO. The set thus contains either one CC or multiple ones, because a KO can be part of more than one CC. The set of CCs that are next when following the macro LP is a logical consequence of the previous

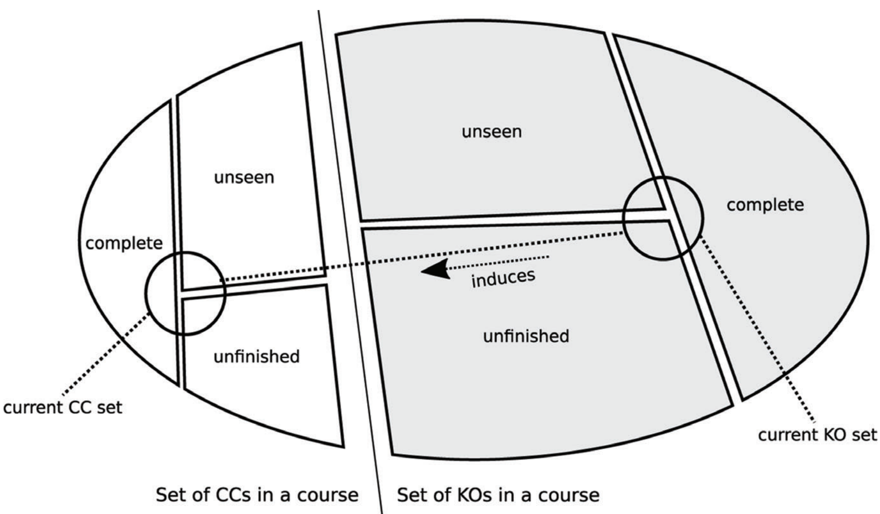

Figure 3.4 Sets of current CCs and KOs in the set of available LOs.

set. When it is known which CC(s) is/are currently active, the next CCs are those that are subsequent as described in the respective LP definition.

With these sets, the back end is able to identify the CCs that have recommendation relevance for the respective learner. This firstly is the intersection of the set of current CCs with the union of unfinished and unseen Ccs. If the set of CCs with recommendation relevance is empty, the course is either complete or there are no more CCs available in that particular macro LP. In the first case, a recommendation cannot be created. In the second case, a new macro LP needs to be selected or the process is finished too.

If at least one recommendation relevant CC has been identified, it is subsequently possible to create a list of KOs that have a CC-based relevance. This is the set of KOs that are attached to this/these particular CC(s). This is already a big reduction of the available LOs, but it can be reduced even further, when also including the micro LP information (Figure 3.5). This is the set that contains all KOs that are "next" as seen from the currently active KO and those KOs that are "next" regarding the "next" CCs. As previously, the term "next" is a matter of definition, which is based on the question what is reasonable to include here.

As the last step in the procedure, to identify those KOs that are recommendation relevant, the back end has to select those KOs that fulfill a list of requirements:

- KO is next or currently active
- KO is attached to a CC that is next or currently active
- KO must be unfinished or unseen (i.e., not already completed)

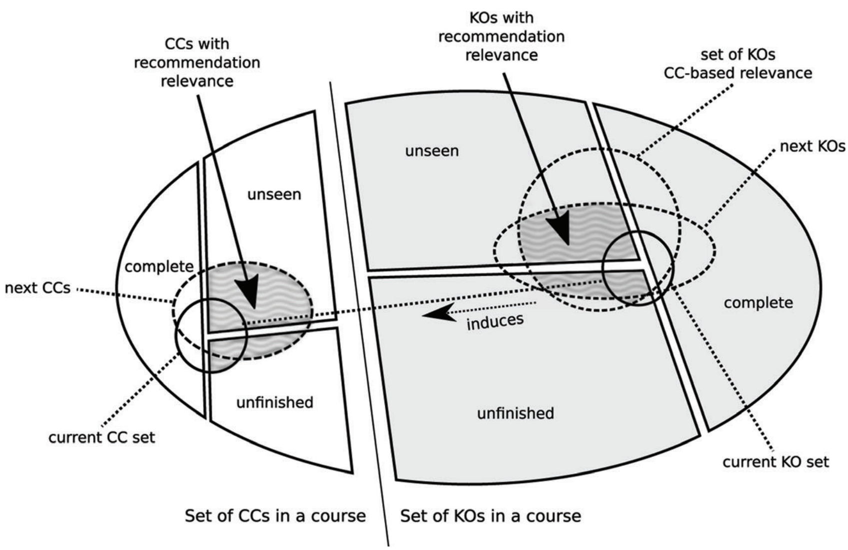

Figure 3.5 Set of KOs that are next regarding the micro LP.

Expressed on a set-basis, this is the intersection of the set of currently active KOs, with the set of KOs with CC-based relevance and the union of unfinished and unseen KOs. For an easier understanding, the diagram in Figure 3.5 clarifies these coherences graphically. If this respective set is empty, the same method as already used for an empty set of CCs with recommendation relevance is applied. Either has the micro and/or macro LP to be changed to include other LOs, or the process is finished, since there are no more KOs available to choose from.

When now coming back to the initial statement that there are three influential factors for the recommendation (macro LP, micro LP and situational dependent information), this so far only includes the first two aspects. In order to further personalize the recommendations the information that INTUITEL collects about the learner must be included in this procedure. For this, the back end specifies a method to include this information in a suitable way, which is implemented via the so called Didactic Factors that are explained in Section 3.2.

On the next level, these Didactic Factors are combined with properties of KOs to state their individual suitability for the learner. The procedure of connecting this information with the learning content is also expressible via the set-based approach as it has been applied to find the (LP-) relevant KOs. Therefore, for each Rating Factor, a set is specified that fulfills a particular rating rule. The task of the back end is then to find the elements that meet the respective conditions and to combine all this knowledge (Figure 3.6). This is done by calculating the intersection of the set of KOs with a general recommendation value (i.e., those elements that have been selected previously on basis of LPs) and those sets that express the optimum regarding the different Rating Factors.

3.3.9 Learning Progress and Learner Position

In INTUITEL the learner is being located in a multidimensional space. This section explains the intentions behind the Multidimensional Cognitive Space (MCS) and the associated Cognitive Content Space (CCS). Both are newly introduced concepts that the INTUITEL project uses to formalize the task of finding the position of the learner in the e-learning content. It has been designed to translate the basic educational conditions to a mathematical level, which is much better applicable on the domain of computer science. The presented geometrical representation of Learning Pathways and learning content in form of a multidimensional hypercube facilitates a better understanding of the technical realization of this task.

3.3.10 Determination of the Next KO

In the following it is outlined how to determine initial recommendations for a “next” KO to be processed by the learner, which are then modified by the Didactic Factors. Note, that throughout this section we are not dealing with Micro Learning Pathways

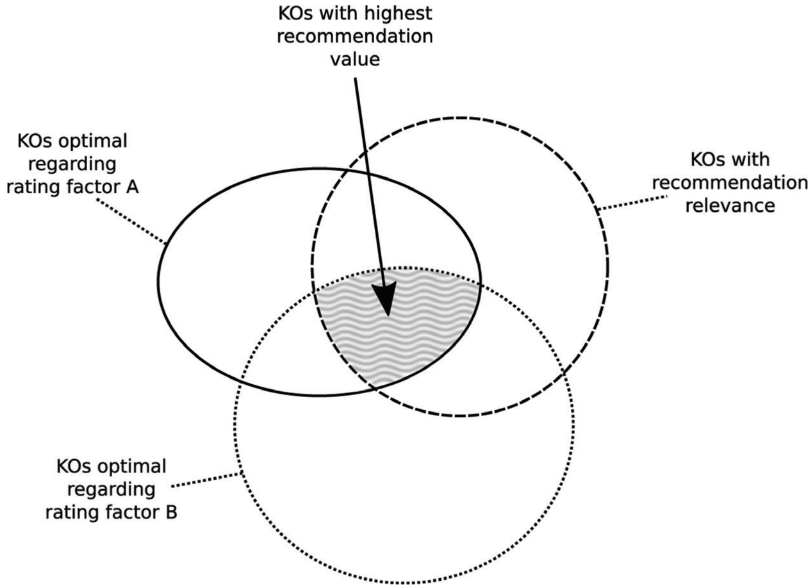

Figure 3.6 Set of KOs with the highest recommendation value.

or LOs that are considered alternatives, but only with Macro Learning Pathways and singular LP nodes. Consequently, for the purpose of this consideration, each KO may be considered as fundamentally different from all the others because it leads to a different knowledge strain. The basic paradigm of outcome-oriented constructivist learning is that it should lead to a certain learning goal. This goal might be quite complex, e.g., might require that

- a certain set of concepts is learned with a required precision
- a certain set of temporal, causal, or logical connections between these concepts is learned
- a certain set of methods and algorithms is learned which enables the learner to process new situations

Mathematically, this complex goal – which in fact is an ontology – may be expressed as a target position in an abstract space spanned by the concepts.

We term this the Multidimensional Cognitive Space MCS. Learning therefore may be considered a movement in this abstract space and learning should bring the learner closer to the target position. In modern human-centered teaching processes, teachers determine the current position of a learner by assessing the learners current knowledge and standing in regard to the topic and on a “meta”-level (e.g., how the learner learns). This happens either subconsciously by sensing that there are problems, delays or disturbances while the lessons are taking place, by evaluating the assignments of the learners (e.g., homework) or by having a direct conversations with them (e.g., questions during the lessons). These observations and feedback enables the teacher to come to conclusions regarding the learners' state of knowledge, habits, and characteristics to ultimately support the learner in the learning process.

INTUITEL provides means to carry out this tutorial guidance process also in technology enhanced learning, where no human teacher in traditional form is present and the learning so far may be seen as a self-directed process. A learning object recommendation, as it is generated by the INTUITEL system, therefore is a computer generated hint towards the self-directed learner on how the learner may come closer to the target position. Note that this recommendation is not entirely a global recommendation ("direct way to the target"), but may be locally deviant from the direct way to the target. Consider, for example, the possibility of a recommendation indicating that the learner should repeat a certain Learning Object: In this case it might correspond to a movement of the learner in the Multidimensional Cognitive Space (MCS) which for some time takes the learner further away from the final learning target position. The INTUITEL system therefore is handling the following tasks:

- Determination of the learner position in MCS
- Determination of a set of Didactic Factors influencing the learning process
- Application of the personal Learning Pathway on the course material
- Taking into account these Didactic Factors and the Learning Pathways, determination of the next step the learner has to take in order to ultimately come closer to the target position

The principles of the MCS will be reconsidered below, for now let us consider the Didactic Factors as they were explained in Section 3.2. Some of them are:

- Cognitive speed, which is the learning speed in the context of environment, personal attitude and current learner position is this a fast, a medium or a slow learner?
- Learning success is this a learner with superior, good, average or inferior results?
- Learning discipline does this learner follow suggestions easily, does the learner adhere to ascribed learning pathways?

For the moment, the INTUITEL consortium relies on a rather simple scale for each of these factors, e.g., "learning speed" is not measured in objective terms but in simple concepts of fast medium slow. This means, that a crucial module of the LPM is responsible to translate raw input data (from the LPM) into these Didactic Factors. Consider, for example the Didactic Factor of "learning speed". For each learner, the LMS delivers the actual learning time for each KO to the LPM. Furthermore, the LPM reads the metadata accompanying the learning content and knows the target learning time entered there by the cognitive engineer (henceforth called "estimated time"). The corresponding transformation rule for the LPM then reads, for example:

> "If the learner needs more time than the estimated time to process a KO in at least 70 percent of all KOs, the learner will be assigned the value "slow" on the dimension of learning speed. If the learner needs less time than the estimated time to process a KO in at least 70 percent of the KOs, the learner will be assigned the value 'fast' on the dimension of learning speed. In any other case, the learner will be assigned the learning speed value "medium" (70-70-rule)."

However, suppose that the learner has already achieved a high level of knowledge, easy questions should then be answered faster. Therefore, one could think to shift from a 70-70-rule for learner speed determination to a 50-85-rule in this case. Or it may turn out that the learner is working in a high noise environment providing lots of distraction in this case one could revert to an 85-50-rule.

3.3.11 Cognitive Content Space

The Cognitive Content Space (CCS) is spanned by the Cognitive Model and the Semantic Learning Object Model (SLOM). SLOM is explained in more detail in Section 3.6. The Cognitive Model contains the pedagogical structure of a course; it is a concretion of the pedagogical ontology for a given domain of knowledge. The SLOM contains meta-data referring to the actual content. Both of these might contribute to the desired learning goal. For later development it is also possible that a more complex learning goal is defined in the Cognitive Model. For example, the cognitive engineer could specify, that in his Cognitive Model the desired learning goal puts more emphasis on practical knowledge than on theoretical knowledge. In such a case, the Cognitive Model would contain a transformation rule to determine the cognitive position from the raw input data.

3.3.12 Multidimensional Cognitive Space

The cognitive position of a learner is estimated by the amount the learner has learned from each of the KOs. Hence, the position for a course consisting of n KOs is determined by an n-dimensional vector within the Multidimensional Cognitive Space (MCS). The value of each vector element can be determined as follows:

- In a simple LMS one only knows that a KO has been processed. In this case we follow the strong optimistic learner assumption: Processing means learning, consequently the grade of progress of the respective KO jumps from 0 to 100 percent when a KO has been accessed.
- A more advanced LMS may tell INTUITEL which part of a KO has been processed by the learner. In this case we will follow the weak optimistic learner assumption: A partial processing of a KO means that the learner has learned the same percentage of this KO.

- A very advanced LMS will tell INTUITEL the result of a measurement, like, e.g., the percentage of points reached in a concluding test of this KO.

Let us now consider how a certain learning goal may be achieved as a sequence of cognitive positions. Let $P = (x_1, x_2, \ldots, xN)$ denote a vector representing the learners position in the Multidimensional Cognitive Space (MCS) with x_i representing the grade of progress of the i^{th} KO.

The default learning goal states, that each KO has to be learned. The target position in the MCS, therefore, is a value of 100% = 1.0 for each component of this vector. The learner starts at position $Ps = (0, 0, \ldots, 0)$, his target position is $Pf = (1, 1, \ldots, 1)$.

Should there be a more complex learning goal (say, target learning profile) instead of all KOs being processed completely, the Cognitive Model defines a transformation, which for simplicity may be seen as a linear transformation (a matrix) reducing the dimensionality of the MCS from N to $M < N$. Therefore, without loss of generality, even in this case the same INTUITEL concepts and algorithms may be used as for the default learning goal.

If we assume a learner who completes each KO before moving on, the movement of this learner in MCS would always be along the edges of a N-dimensional hypercube: The learner has completed KOs 1, 2, 3, 4, . . ., $k - 1$. He is currently working himself through the k^{th} KO and still has to consider himself with KOs $k + 1$, $k + 2$, ..., N. Hence, his cognitive position would be

$$Lk = (1,\ 1,\ 1, ..., x_k, 0,\ 0, ..., 0)$$

Whereby x_k represents the KO at the current position. Here, without loss of generality, we have ordered the KOs in exactly the sequence as it is processed by the Learner, henceforth called User Learning Path (ULP). However, we might also assume a learner who is much less disciplined, and therefore does not complete any KO before moving on. Obviously, the sequence of cognitive positions in the MCS would be a curve inside (and not along the edges, but possibly across bounding surfaces) of the hypercube. In the extreme example, this learner would switch back and forth between the KOs in the model and finish each KO to the same degree. While this learner might nevertheless reach the final learning goal, his actual learning curve is a line close to the hyper-diagonal of the MCS. In order to achieve his final goal however, this learner would have to perform a very large number of switches between KOs, in order not to emphasize a particular KO.

For now let us assume that we have a disciplined learner who always moves along the edges of the knowledge hypercube. This amounts to a self-chosen sequence of KOs, each of them is processed (by assumption, learned) completely before moving on. After completing the k_{th} KO, the cognitive position would be

$$L_k = (1, 1, 1, \ldots, 1, 0, 0, \ldots, 0)$$

Obviously, if repetitions are excluded, this learner then has $N - k$ possible choices for his next KO and the direct cognitive distance to the target position is $\sqrt{N - k}$.

Each of those possible steps would reduce the direct cognitive distance by the same amount. Hence, none of the remaining KOs would be preferable from the viewpoint of reducing direct cognitive distance to the target state.

The Cognitive Model however may define certain Learning Pathways $LP_1, LP_2, \ldots, LP_S$. Since we reserved the standard numbering for the ULP, we have to consider each of these predefined Learning Pathways an ordered permutation of the numbers $1, 2, \ldots, N$. Since the current cognitive position after k learning steps may be arbitrarily close to Learning Pathway l but after a number of t steps we may not use simple path deformation rules to compare the actual current cognitive position to one that could be reached by this Learning Pathway. Therefore the LPM implements the following algorithm:

1. Determine the Learning Pathway LP_l from the Cognitive Model which runs closest (in distance d) to the current cognitive position L, and the closest corner point V (vertex) of P_l
2. Check each of the open choices $N - k$, if it would reduce the distance to V from d to $d' < d$ and assign to it the priority $d - d'$.
3. Check on the Learning Pathway P_l which the next KO would be after V, we assume that this would be KO no. v. If v is among the open choices $N - k$, add to the priority assignment of v the value $+1$ and terminate the loop. If not, choose the successor of v in P_l and check if it is among the open choice, add to its priority the value of 0.5, etc. Continue if either this loop terminates or if the end of P_l is reached.
4. Pass the information to the next decision stage.

The current cognitive position L is close to P_l and the next object recommended when P_l is followed by the one of the $N-k$ open choices which has the highest cumulated priority. If there is more than one next choice with the same priority, take a random selection among those. This algorithm ensures that a recommendation to the learner will, if the learner follows the recommendation, propagate through the MCS roughly in parallel to a certain Learning Pathway and at the same time towards this particular Learning Pathway. It is possible to relax this and present to the next higher decision stage a selection of alternative Learning Pathways and the corresponding suggestion for the next KO.

3.3.13 Example for Learner Positions

We now consider a system consisting of four different KOs A, B, C and D. The Multidimensional Cognitive Space then is a four-dimensional hypercube or tesseract. A three-dimensional projection of this hypercube is depicted in Figure 3.7.

The cognitive position of a learner is determined by the amount the learner has learned from each of these four KOs. Hence, the position is determined by a four-dimensional vector within the hypercube. The learner starts at position $P\text{s} = (0, 0, 0, 0)$, which is the lower left of the inner cube of the above projection. His target position is

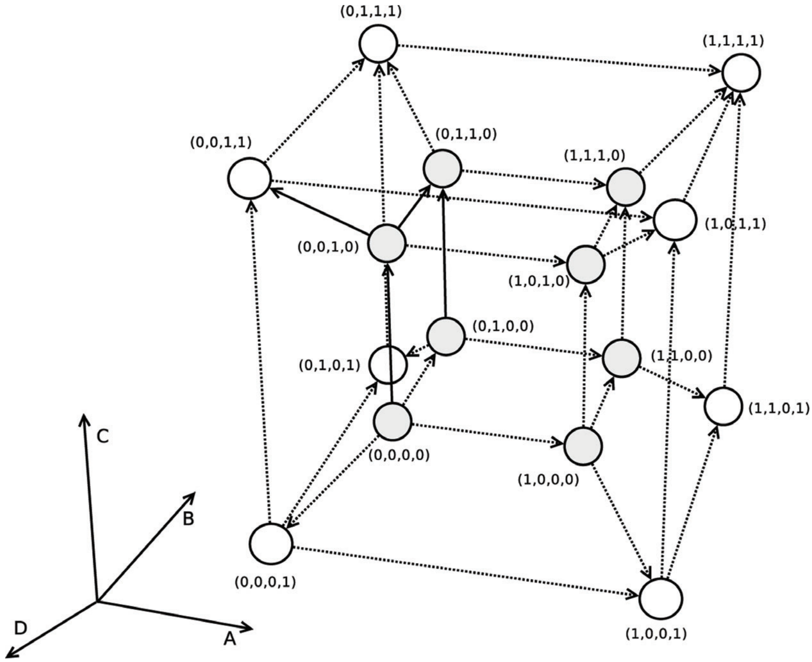

Figure 3.7 Four-dimensional cognitive space (4D hypercube) for four KOs.

Pf = (1, 1, 1, 1), which corresponds to the upper right of the outer cube of the above projection. In principle 4! = 24 distinct Learning Pathways exist and therefore, since the learner is free to choose, may be transgressed by the learner. Not all of these are didactically meaningful; hence let us assume that the KOs are grouped:

- The two pairs A, B, resp., C, D each constitute a thematic group.
- The pairs A, C, resp., B, D each constitute a chronological group. Such a grouping could arise, if
- A describes the contribution of Comenius to educational theory
- B describes the contribution of Habermas to educational theory
- C describes the contribution of Comenius to communicative didactics
- D describes the contribution of Habermas to communicative didactics

Consequently, two Learning pathways are defined in the Cognitive Model:

$$LP_h = (A, B, C, D) = (K_1, K_2, K_3, K_4) \text{ as } \textit{hierarchical} \text{ LP}$$
$$LP_c = (A, C, B, D) = (K_1, K_3, K_2, K_4) \text{ as } \textit{chronological} \text{ LP}$$

These two LPs differ only in one permutation of K_2, K_3. They are depicted by the thick arrows in the projection in Figure 3.8

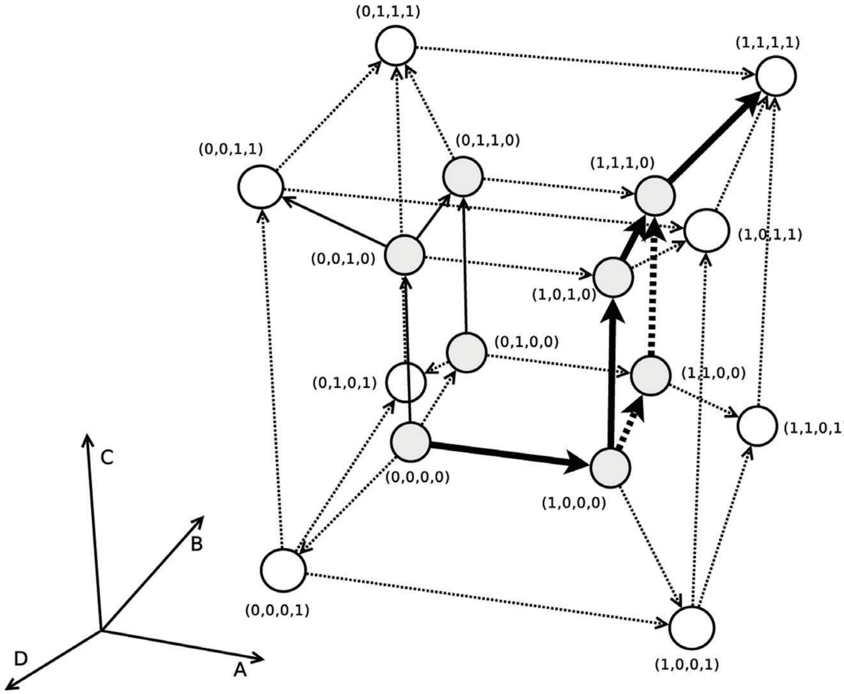

Figure 3.8 The two didactically meaningful learning pathways (hierarchical and chronological), shown as thick arrows.

Let us now follow a learner through this system. First of all let us assume a disciplined learner: The learner follows the initial advice and processes KO A, which puts the learner into cognitive position (1, 0, 0, 0). Obviously, the learner is still following both LPs defined in the Cognitive Model. If no other factor (like a pre-chosen LP) will lead the INTUITEL Engine to another conclusion, the learner will be offered two choices by the INTUITEL system:

> "If you want to follow the hierarchical LP, please process B. If you want to follow the chronological LP, please process C."

Conversely, we assume an undisciplined learner who, instead of following the initial advice to process A, has chosen to process D instead. This would put the learner after the first learning step into cognitive position (0, 0, 0, 1). Using the previously described algorithm, the LPM would determine that the learner is close to position (0, 0, 0, 0) which is on both well-defined LP. The learner would then get the advice to process A as the next KO, bringing the learner (if the advice is followed) into position (1, 0, 0, 1). There, the learner is close to position (1, 0, 0, 0), which is on both LP. Therefore, with equal priority (unless other factors come into play) the learner will be advised to process B or C. Suppose B is followed, this would put the learner into position (1, 1, 0, 1) where the learner would finally get the advice to process C leading to the learning goal. The actual LP then is (*D, A, B, C*) where at least three KO are in the sequence which has been termed adequate by the cognitive engineer.

However, our learner might be so undisciplined that instead of following the first advice the learner takes a complete different route processing C after having done the first step of processing D. This would take the learner into position (0, 0, 1, 1) and the closest of the well-defined LP's is the chronological LP with position (1, 0, 1, 0). The learner would therefore get the advice "You are close to the chronological LP, please process B". If the learner follows this advice, the learner is led into position (0, 1, 1, 1) where then the learner would get the final advice to process A and therefore the overall realized LP would be (*D, C, B, A*). In this realization, only the two KOs C and B are in the chronological sequence described in the Cognitive Model. We might also encounter a totally undisciplined learner, who also does not follow the second advice, e.g., processing first D, then C and then A. This would again bring the learner close to the chronological LP, and the learner would receive the final advice to process B. The actual LP then is (*D, C, A, B*) whereas in the previous scenario, only the two KOs A, B are in the chronological sequence described by the Cognitive Model.

3.3.14 Learner-State Ontology: The Output of the LPM

The output of the LPM consists of two parts. Firstly, there is the output of the actual task that the LPM carries out, the set of applicable Didactic Factors. These are defined in context of the Learning Model Ontology as elaborated in Sections 3.1 and 3.2. These Didactic Factors are essential for the individual learner-related rating of learning objects. The LPM prepares the input and particularly prepares the Didactic Factors that are passed to the Engine. It, for instance, checks the connectivity of the learner and ranks it, to state that the learner has a good or bad connection. This represents the learner-specific knowledge of the reasoner and allows the Engine to include this information in the deductive process.

However, the set of actually relevant Didactic Factors alone is not sufficient for the INTUITEL Engine. An OWL reasoner needs an ontology to perform its work and therefore, the LPM must create one that contains all relevant data, i.e., the second part of the LPM output. This basically is a merge of those metadata that describe the respective course (including its Cognitive Model), the micro Learning Pathway information from the Pedagogical Ontology and the Learning Model Ontology, containing the machine-readable descriptions of Didactic Factors. To combine this information, the LPM creates the so-called Learner-StateOntology. This only temporarily valid collection depicts the current status of the learner in that course with respect to his or her learning history and current environment. Figure 3.9 illustrates the parts that are merged into the Learner State Ontology as the final output of the LPM.

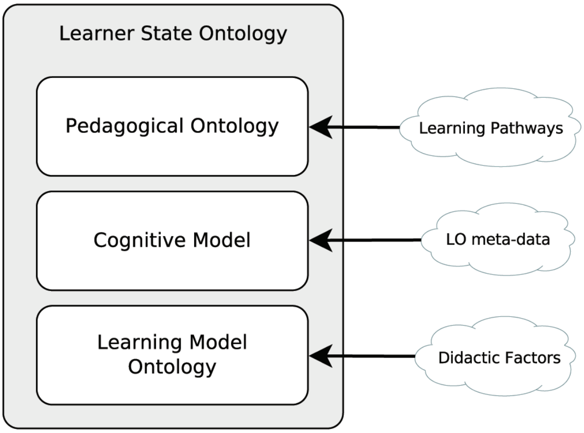

Figure 3.9 The final output of the LPM: the Learner State Ontology.