Deliverable D3.4

Third Release of LA/EDM Services and Algorithms

Document Version:	v1.2 – 12 October 2016
Document Status:	Final
Document Type:	Report
Diss. Level:	Public
Lead Partner:	TUGraz
Delivery Date:	M31 (September 2016)

Authors: Michael Kickmeier-Rust,
Kolja Debus,
Michael Bedek

TABLE OF CONTENTS

1. Introduction

One of the main goals of the project was to conduct research with the goal to apply the theories of Knowledge Space Theory (KST), Competence-based Knowledge Space Theory (CbKST), and Formal Concept Analysis (FCA) to the field of Learning Analytics and Educational Data Mining.

This work has two strands, the one strand is to apply the principles and solutions, transform, and adjust them to the needs of Learning Analytics. These needs are quite different from the origin of the theories, which is autonomous, intelligent and adaptive tutoring. This means that the original focus was on equipping software system with an understanding of human learning activities and learning progress. The focus of Learning Analytics, however, is to better inform human teachers about learning processes. Thus, the principles and solutions could not be translated one on one.

In the past periods we have described and reported on the work accomplished in this transformation context in detail. In addition to these achievements, we perused the second strand, conducting fundamental theoretical research to advance the state of the art by bringing the (in fact disconnected) theories together. These endeavors lead to a final expert workshop with leading experts in this filed finalizing the research work of the project and describe the achievements for the entire community (we will report about this event in the context of WP6).

The research work was done together with leading experts and led to draft research papers, which are attached to this document. The following section, briefly summarizes, once again, the approach and solutions.

1.1 Knowledge Space Theory and its Competence-based Extensions

The Knowledge Space Theory (KST, Doignon and Falmagne, 1985) suggests that every knowledge domain *Q* (e.g. arithmetic) can be characterized by a set of problems. In this context, problems should be considered as “type of problems” consisting of concrete instantiations in the form of test items. The set of problems a student is capable to master is called his or her knowledge state. In many domains it is reasonable to assume mutual dependencies, so-called prerequisites between the problems. For example, a student who successfully masters problem *y* (e.g. multiplication of single-digit numbers) presumably masters problem *x* (e.g. addition of single-digit numbers) too. In this case problem *x* is called "a prerequisite" of problem *y* (also denoted as $x \leq y$). The set of problems together with its prerequisite relation (Q, ≤) is a partially ordered set and can be represented as a Hasse-diagram (see Figure 1, left). Every knowledge state which includes a particular problem also encompasses the problem´s prerequisites. The set of knowledge states is called a knowledge structure. A knowledge structure which is closed under union is called a knowledge space (Doignon and Falmagne, 1999).

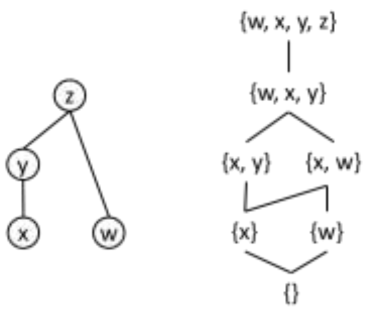

Figure 1. A partially ordered set (Hasse-diagram) shows the prerequisite relation on the set of problems (left) which defines the knowledge structure, the ordered set of plausible knowledge states (right)

As a behaviouristic theory, which in fact means promoting a superficial, perhaps naïve view on activities exhibited in specific scenarios or environments (e.g. an online quiz), the KST focuses solely on observations and does not address the imperative question of why a student masters a problem, task or item and why not. There might be an array of reasons ranging from lucky guessing to careless errors, or from the presence of skills and competences to conceptual misconceptions. This is addressed by CbKST (e.g. Korossy, 1999; Heller et al., 2006). The CbKST focuses on the underlying cognitive constructs such as competences (or skills, abilities or entities of aptitude) which enable students to master the problems of the given domain. In a nutshell, CbKST extends the knowledge space with an additional space, the competence space which is the ordered set of competence states. Analogously as for the knowledge states, the competence states are determined by prerequisite relations among the competences. The connection between these two spaces is ensured by a skill function that associates to each problem those competence states that are sufficient for solving it (Heller et al., 2013).

This competence-centred extension has two main advantages: First, given the performance, i.e. the student´s knowledge state, the latent underlying competences can be identified. Second - as it will be further outlined in the subsection on providing feedback to students below - pedagogically-sound competence development can be realized by individual learning paths through the competence structure. In other words, CbKST's competence spaces provides a solid scaffolding to understand an individual's competencies, competency gaps, learning processes, and learning sequences and it allows making a cautious link of behaviours and performances exhibited in various situations or environments.

AN EXAMPLE

Several methods to elaborate knowledge structures and knowledge spaces have been suggested, for example by systematic problem construction (e.g. Albert & Held, 1994), by querying experts (e.g. Düntsch & Gediga, 1996; Koppen, 1994) or by analysing empirical solution patterns by a sample of participants (e.g. Schrepp, 2003, see also section 0). A sophisticated approach is based on an in-depth analysis of the problems / tasks, the required solution methods and underlying "elementary competencies" and the functions between subsets of problems and elementary competencies and vice versa (e.g. Korossy, 1999). From this perspective, the approach from Korossy is a purely "theory-driven" methodology for constructing knowledge spaces. An example knowledge space, representing the dependencies of 6 problems in the domain elementary algebra is shown in Figure 2. Finally, the theoretically deducted knowledge space has to be validated by comparing the theoretical structure with the empirical data (Korossy, 1997).

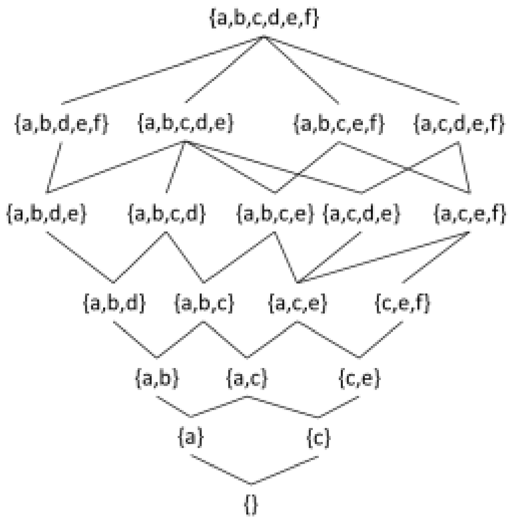

Figure 2. A knowledge space represents the dependies between 6 items (based on Korossy, 1999)

1.2. Formal Concept Analysis

The FCA (Wille, 1982; 2005) aims to describe concepts and concept hierarchies in mathematical terms. The starting point is the definition of a formal context *K*, which is a triple (*G*, *M*, *I*) with *G* as a set of objects and *M* as a set of attributes and finally, *I* as an incidence-relation which assigns objects to attributes and vice versa. The formal context can be represented as a cross table, with the objects in the rows, the attributes in the columns and by crosses ("Xs") in the cells whenever g*I*m holds for a particular object and attribute. Table 1 shows an example of a formal context.

Table 1. Example of a formal context with objects, attributes, and the incidence-relation

Objects *G* / Attributes *M*	is toxic	is able to fly	is able to swim	hatched from egg
Bee	X	X		X
Bumble-bee		X		X
Tree frog			X	X
Grass snake	X		X	X

For each subset *A* of objects and each subset *B* of attributes, the so called derivations $A \rightarrow A'$ and $B \rightarrow B'$ can be defined as follows: A' is the set of attributes which are common to all objects in *A* and B' is the set of objects which share all attributes in *B*. A formal concept is a pair (*A*, *B*) with the subsets $A \subseteq G$ and $B \subseteq M$ which fulfil $A' = B$ and $B' = A$. The set of objects *A* is called the *extension* of the formal concept. The set of attributes *B* is called the *intension* of the formal concept. A formal concept (A_1, B_1) is a sub-concept of the concept (A_2, B_2) if $A_1 \subseteq A_2$ (which is equivalent to $B_1 \subseteq B_2$). The set of all formal concepts which is ordered by such a sub-superconcept relation is called concept lattice *B*(*K*) (see Wille, 2005). A lattice is an algebraic structure whereas for each pair of elements there exists a unique supremum and a unique infimum (for further details see Davey and Priestley, 2002). The concept lattice *B*(*K*) can be visualized as a labelled line diagram (see Figure 2).

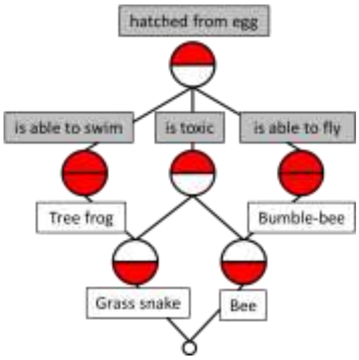

Figure 3. A concept lattice resulting from the formal context in table 1.

Every node of the concept lattice is a formal concept. The domains´ attributes and objects are labelled only once to avoid redundancy. The concept lattice can be "read" as follows: The extension *A* comprises all objects whose labels can be reached by descending paths. As an example, the formal concept labelled with "Tree frog" has the extension {Tree frog, Grass snake}. The intension *B* comprises all attributes whose labels can be reached by ascending paths from that node. In the example above, the intension consists of the attributes {hatched from egg, is able to swim}.

AN EXAMPLE

Rusch and Wille (1996) were the first who applied the FCA with students and their knowledge states, aiming to show the correspondence between FCA and KST from a mathematical point of view. They proposed a knowledge context (*S*, *P*, *I*) with students *S* as objects, problems *P* as attributes and an incidence-relation that maps students to problems which they have *not solved*. Such an incidence relation leads to formal concepts *whose complements of the intensions are knowledge states*.

However, such kind of knowledge contexts and concept lattices are hardly applicable since it is not intuitive to think in terms of "complements of a formal concept´s intension". Easier to read concept lattice result when student are represented as "attributes" and problems as "objects" and an incidence relation which means "student has solved problem". An example of such a transposed knowledge context is given in table 2 (data reported by Korrossy, 1999).

Table 2. A knowledge context with student as attributes and problems as objects (from Korossy, 1999)

Students Problem	*01*	*02*	*03*	*04*	*05*	*06*	*07*	*08*	*09*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*
a	X	X	X	X	X	X		X	X		X	X	X	X	X	X	X	X	X	X	X	X	X
b		X	X	X	X		X		X		X	X		X	X		X			X	X		
c	X	X	X		X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
d	X		X		X		X										X						
e	X	X	X		X		X				X	X	X				X						X
f	X	X	X				X						X		X		X						X

Transposed knowledge contexts lead to formal concepts whereas a student´s knowledge state can be directly derived from the according concept´s extension (compare left and right side of Figure 3). In addition to that, as it will be outlined in the following sections, the resulting concept lattice allows visualizing answers to a set of pedagogical questions which are of interest for teachers (see Bedek, Kickmeier-Rust et al, 2015). “Reading” and interpreting such concept lattices requires a certain level of training, which is also true for other kinds of hierarchical graphs (Körner, 2005). As for example described by Spangenberg and Wolff (1991) who used concept lattices for the evaluation of psychoanalytic data, there can be steep learning curves for users of concept lattices. However, once understood, the lattices bear highly important conclusions that not necessarily can be drawn from simpler approaches to learning analytics.

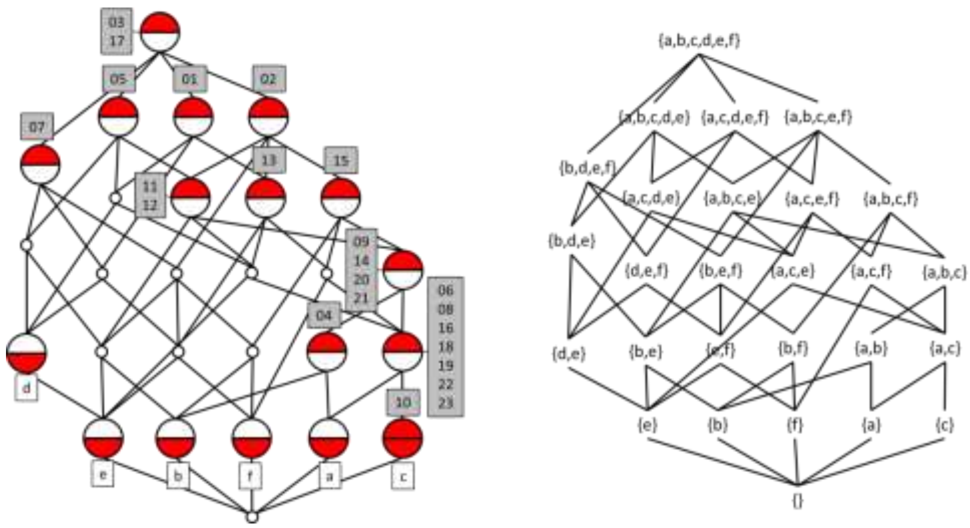

Figure 4. Left: Concept lattice with students as attributes (numbers from 01 to 23) and problems (test items; letters a, b, c, d, e, and f) as objects (data reported by Korossy, 1999). Right: The extensions of the formal concepts represent knowledge states in the knowledge structure

1.3. DEPICTING INFORMATION FROM THE FORMAL CONCEPTS EXTENSIONS AND INTENSIONS

As mentioned above, the formal concepts' extensions reflect – either empirically or potentially - knowledge states. As an example, student *04* solved the problems *a* and *b* (see Figure 4). Better performing students are located above lower performing students in the concept lattice.

The intension of a formal concept, which has an object-label assigned to it, indicates these students who have successfully mastered the according problem. As an example, the problem *d* in Figure 4 has been mastered by the students *01*, *03*, *05*, *07* and *17*.

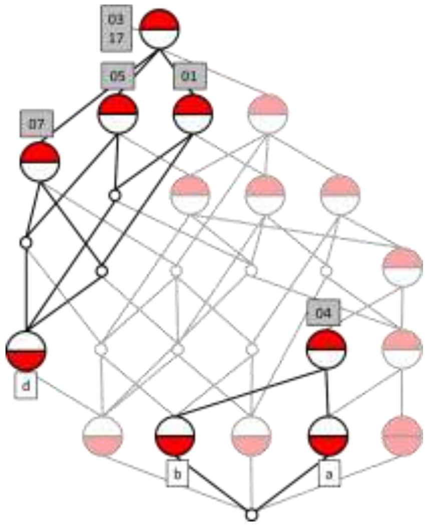

Figure 5. The extension represents the set of test items solved by a student (see student *04*) and the intension indicates the students who mastered the particular test item (students *01*, *03*, *05*, *07* and *17* test item *d*)

1.4. Inductive Item Tree Analysis

The *Inductive Item Tree Analysis* (IITA, Schrepp, 2003, 2006) is a method which aims to derive hierarchical dependencies between the items of a questionnaire. The underlying idea and its main principles of the IITA originate from the ("Classical") Item Tree Analysis (CITA, Van Leuwe, 1974). In general, the items (problems, tasks) have to be continuous, i.e. the subjects can either agree (1) or disagree (0), or the response to the problem or task can be either correct (1) or incorrect (0).

Based on the observed response patterns of *n* participants to *m* items (which can be represented as a binary matrix *D* with *m* columns and *n* rows), the IITA derives dependencies in the form "item *j* implies item *i"* (denoted as $j \rightarrow i$). "Item *j* implies item *i"* means that we can surmise from a correct (or "agreed") response to item *j* a correct (or "agreed") response to item *i.* Thus, the IITA can be seen as a data-driven (rather than theory driven approach as described in section 1.1) for defining knowledge structures and knowledge spaces.

In the following, we will describe the principles and steps of the IITA on a rather surface level (for details see Schrepp, 2003). The according software tool and it´s features have been described by Schrepp (2006).

In a first step, a so-called b*ij* matrix which represents the b*ij* values for each pair of items has to be established. The b*ij* value for the items *i* and *j* is the number of counterexamples for the implication $j \rightarrow i$. As mentioned above, $j \rightarrow i$ means that we can surmise from a correct (or "agreed") response to item *j* a correct (or "agreed") response to item *i*. Counterexample of such an implication are those cases where a subject mastered item *j* and failed to item *i*. The b*ij* matrix for the response patterns from

Table 2 is shown in the following Table 3.

Table 3. The b*ij* matrix based on the observed response patterns in Table 2.

bij	a	b	c	d	e	f
a	0	1	1	1	1	1
b	9	0	10	1	3	3
c	1	2	0	1	1	1
d	17	9	17	0	5	4
e	12	8	10	0	0	1
f	14	8	14	1	3	0

As an example, the number of counterexamples for the implication the $c \rightarrow d$ is 17 (i.e. b*dc* = 17), which means that 17 out of 23 participants correctly solved item *c* and failed to item *d*. On the other hand, the number of counterexamples for the implication the $d \rightarrow c$ is 1 (i.e. b*cd* = 1), which only 1 out of 23 participants mastered item *d* but failed to item *c.* The remaining 5 out of 23 participants either mastered both items or failed to both items.

Even if there is a single counterexample to $d \rightarrow c$, based on the amount of participants (*n* =23) it reasonable to take this implication as granted. The (Classical) Item Tree Analysis would consider only those implications for which no counterexample exist. However, in large data sets it is rather unlikely to detect such deterministic implications. Compared to this, the IITA can be considered as a more probabilistic since it also considers implications for which (few) counterexamples exist. In other words: Unless the number counterexamples do not exceed a certain threshold, the according implications are considered as valid.

The next step of the IITA is to identify such a threshold and the underlying principle for this is as follows: The goal of the IITA is to uncover the relation ≤ (a relation in the mathematical sense is a set of pairs of elements - in this case the elements are items - for which the relationship holds). The relation ≤ defined upon the set of items (Q, ≤) is a quasi-ordered set. As for partially-ordered sets, quasi-ordered sets and can be represented as a Hasse-diagram (see Figure 1).

The final goal is to identify the “best-fitting” quasi-order, i.e. the quasi-order which represents the response patterns best.

In an iterative process, those items pairs with low b*ij* values are included into the quasi-order, starting with the item-pairs with the lowest b*ij* value. In our example, the lowest b*ij* value is 0 for the item-pair (d, e). Thus, the first quasi-order would consist only of the "implies"-relation (d, e) and the reflexive Itempairs {(a, a), (b, b), (c, c), (d, d), (e, e), (f, f)} on the set of items {a, b, c, d, e, f}. After the identification of the quasi-order including only those item pairs with the lowest b*ij* values, the reproducibility coefficient has to be calculated: the reproducibility coefficient is a particular goodness-of-fit indicator, providing a numeric value for how well the model (the quasi-order) represents the raw data. The reproducibility coefficient represents the relative number of cells whose binary values can be reproduced by the particular quasi-order. As an example: a reproducibility coefficient of 0.91 means that 91% of the cells´ values can be correctly reproduced by the particular quasi-order.

In the next step, the quasi-order including the item-pairs with the lowest as well as the second-lowest b*ij* values has to be identified. In our example, this would include the Item-pairs {(b, a), (c, a), (d, a), (e, a), (f, a), (d, b), (a, c), (d, c), (e, c), (f, c), (f, e), (d, f)} since they have b*ij* values of 1 in addition to the previously identified Item-pairs with a b*ij* value of 0. Again, the reproducibility coefficient of this particular quasi-order has to be computed. This step continues several times (As a rule of thumb, the distribution coefficients is L-shaped, meaning that the value increases with some of the lower b*ij* values and then rapidly decrease with larger b*ij* values).

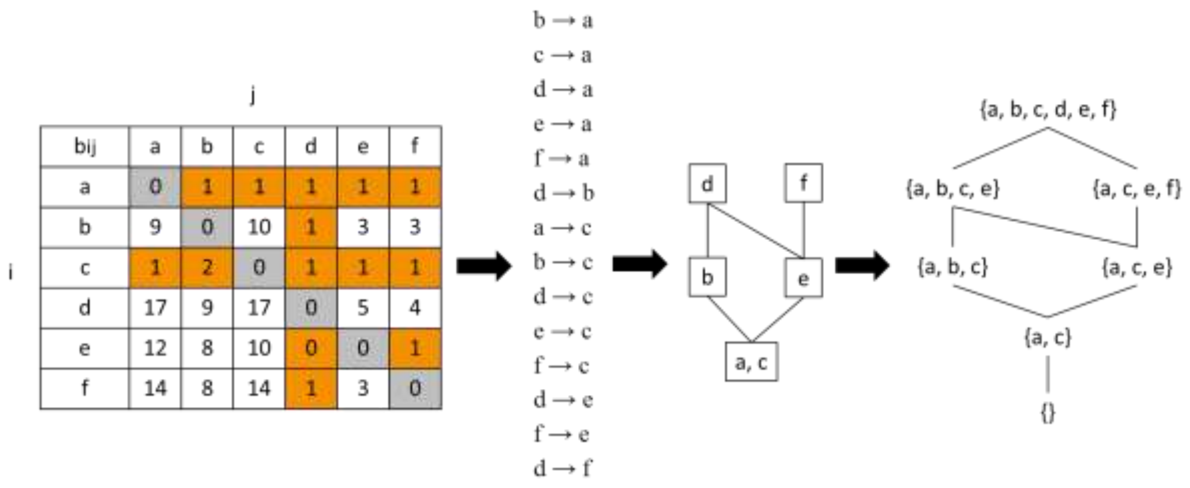

Figure 6. The different steps of the IITA, from a b*ij* matrix (left) to a knowledge space (right)

The quasi-order with the highest reproducibility coefficient will be selected. As indicated in Figure 6, the set of item-pairs can be represented as a Hass-diagram and finally, the Hasse-diagram can be transformed into a knowledge space.

1.5. Comparison between KST, FCA and IITA

As indicated in Figure 7, the different methods described in the previous sections, KST, FCA and IITA, deliver different resulting knowledge structures. The KST can be considered as "theory-driven", the structure and underlying interdependencies between the items or problems are either based on expert ratings or an in-depth analysis of the items or problems, the required solution methods and the relations within and functions between those two sets. It is the most time consuming method. However, compared to the other two methods, the FCA and the IITA, which can be considered as purely "data-driven" approaches, there are some severe advantages: The data-driven approaches are – as the adjective suggests – purely data-driven, i.e. the resulting structures and interdependencies are purely dependent on the available data (or the according sample). On order to be able to generalize the results, the structures need to be validated (e.g. by cross validation). The resulting structures derived from the data-driven approaches do not tell us the underlying reasons for the interdependencies and there is the danger to interpret the structures and to elaborate post-hoc hypothesises about the relationships between items, problems, skills or competences.

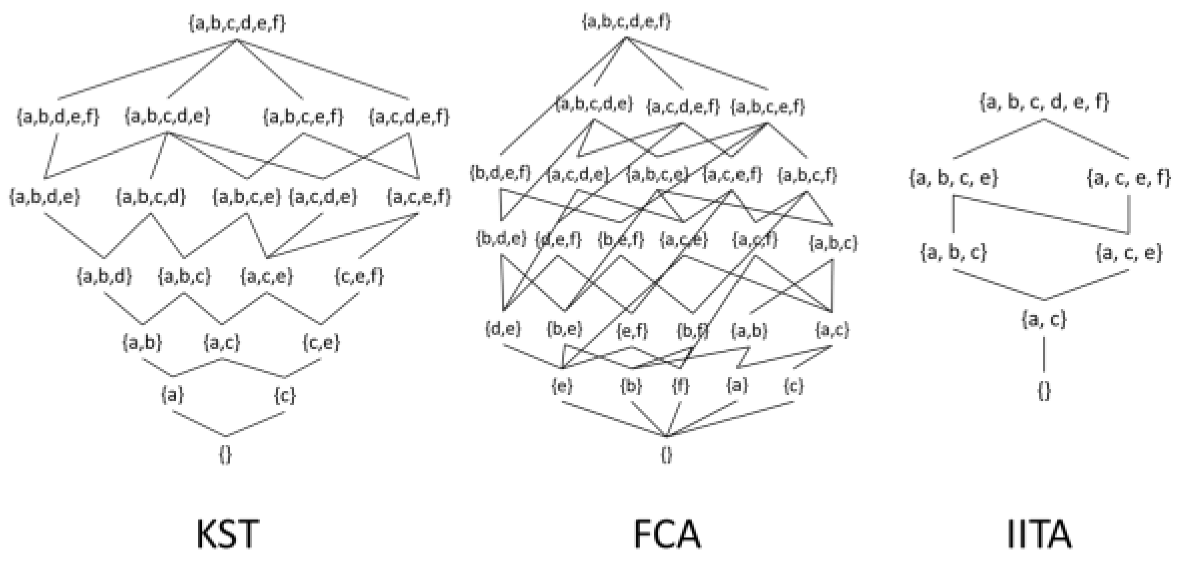

Figure 7. Knowledge Spaces derived from three methods (KST, FCA and IITA) for the same dataset (Korossy, 1999)

In the case of the KST, the assumed interrelationships can be considered as ad-hoc hypothesises and these hypothesises are tested by comparing the theoretical structure with the empirical structure(s). However, in some cases, in particular when neither time nor resources are available, one could consider to apply a “hybrid” approach: applying a data-driven approach in a first step to exploratory collect potential hypothesises, establishing a structure by incorporating these hypothesises in a second step and finally, testing the theoretical structure against empirical data from another sample.

The FCA seems to be too strict on identifying interrelationships between the items – the identified relation between the item-pair (d, e) in the sense $d \rightarrow e$ (*d* implies *e*) or $e \leq d$ (*e* is a prerequisite of *d*) holds for all subjects. In other words, applying the FCA as such an exploratory, data-driven approach to identify interrelationships between the items would not count for careless errors or lucky guesses. From this point of view, it delivers the same results as the CITA. However, the strengths of the FCA are that it enables to answer a broad set of pedagogically relevant questions by visualizing student’s performances on the classroom level without loss of information. Besides that, also other pairs of subsets can be assigned to each other and visualized, such as the competences X students, the competences X activities or the students X activities lattices in the case of the LEA´s BOX FCA-tool.

2. The Learning Performance Vector and the Learning Horizon

A major achievement within Lea’s Box, although not exactly planned in this way, was the Learning Performance Vector (LPV), This vector is a method to predict a student’s learning progress and the chances to reach a general learning goal within a given timespan (e.g., the duration of a course).

The origin of the prediction algorithm is a competence structure (or competence space). This structure gives us a model of the learning domain, starting from point 0 (in this particular domain) leading to the complete mastery. In other terms, a competence structures is the manifestation of all possible and reasonable states a person can be in. This allows us to identify the progress of a particular learner given the timeline of a course. Mathematically speaking we have the sum of all possible learning paths. This indicates the average learning efforts, given that transitions have specific difficulties or weights (as explained above).

We have a set of competencies Q = {a, b, c, …. } with a relationship c ≥ c' among the competencies, which establishes the competence structure. The sum of the resulting competence states is $\Sigma(|Q|r)$. Given that the transitions from one competence state to another has a difficulty parameter, which in turn is the average of the difficulty parameters of the competencies being a part of the state, we have a set of tuples of the start competence state, the end state, and the difficulty τ = [s1, s2, w]. This results in a set of such tuples for the entire competence structure $\mathrm{T} = \Sigma(\tau|Q)$. Also, we have a set of indicators providing evidences for competencies: I = { ei, {c} * w}, with a given weight w.

Based on the evidences we can estimate the likelihood of each competency. The probability of a competence state is the average of its competencies $\Pi(s) = \Sigma(\pi)/n$.

To identify the learning path of a person, we identify the state with the highest probability in certain time steps. Depending on the nature of the concrete use case this may rely on the events when evidences are put into the system or, alternatively on a timely basis (e.g., weekly or monthly). This is basically illustrated in the next figure.

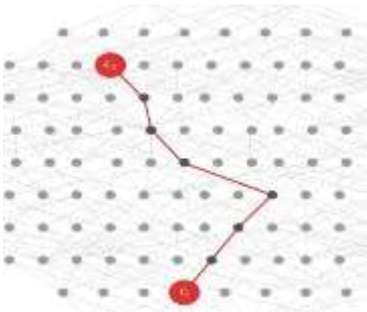

Figure 8. Learning Path. The cutout is part of the structure shown above.

Now for each step we compute the difficulty (as a value from 0 to 1). The sum of the values gives us an indicator for how many efforts a student has to spend on her learning history (the individual learning path). In a next step, given the concrete competence state of the learner, we have to identify the possible paths towards to defined learning goal, which is a (rather small) subset of all possible paths. Equally to the computation of the difficulty to reach the current state, we can compute the potential difficulty of all possible paths to the goal, whereas we have to compute the average difficulty of all possible paths. This now is an indicator for the efforts that are necessary for an individual learner to reach the learning goal.

As indicated in the following figure, when link the progress of a student within a given span of time, we can make a prediction about how far a student can come within the remaining time (of a course, for example). So, as a final step, we can identify exactly those states (and therefore the competencies) a particular will be able to reach within the time limits. The set of those states is, now finally, the student's Learning Horizon.

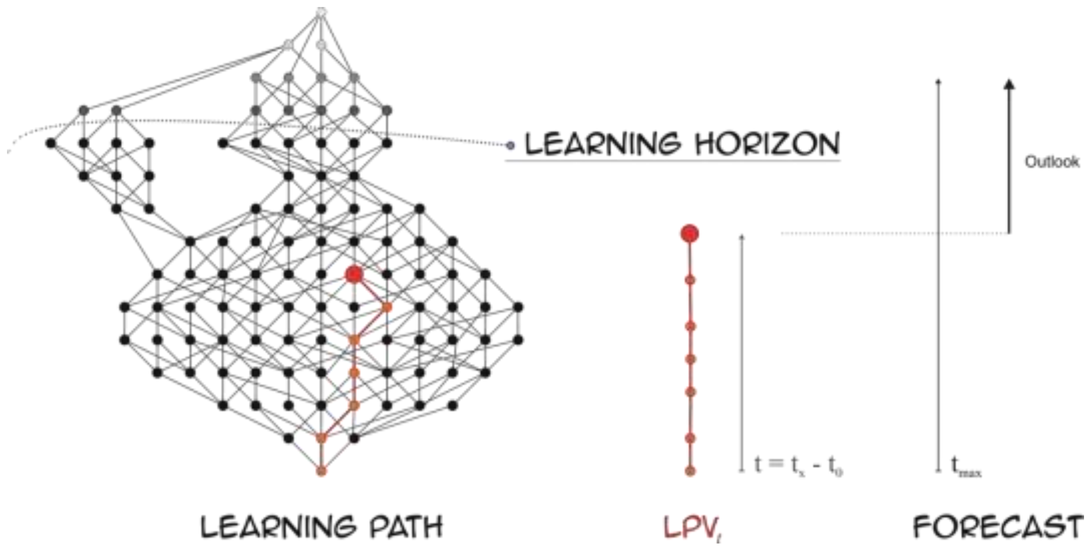

Figure 9. Conceptual sketch of the LPV / LH

While the conceptual and theoretical work has been accomplished in year 2, in the final period we fully implemented the algorithm in the Lea's Box system (as reported in detail in deliverable D2.6). In the remaining months, this prediction method will be evaluated using large data sets from within and outside the project.

3. CbKST Analyses and Hasse Diagram Visualizations

The nature and the intentions of CbKST/FCA-based Learning Analytics have been reported extensively in the previous periods. The manifestation of the analyses is, (most often) a Hasse diagram which is a mathematical representation of a so-called semi-order which helps for structuring learning domains and for visualizing the progress of a learner through this domain. The properties of a semi-order are: (i) reflexivity, (ii) anti symmetry, and (iii) transitivity. The representation of this diagram is illustrated in the image below. The direction of a graph reads from bottom to top. The arrows from one element to itself (reflexivity property) as well as all arrows indicating transitivity are not shown, but they are included (used) so far. In an educational context, a Hasse diagram can display the non-linear path through a learning domain starting from an origin at the beginning of an educational episode (which may be a single school, lesson or the entire semester). The beginning is shown as a {0} (empty set) at the bottom of the diagram. Now a learner might focus on three topics (X, Y, Z). In essence this establishes three possible learning paths, until reaching the final state (X, Y, Z). In the context of formative learning analytics, a competence-oriented approach is necessary. Thus, a Hasse diagram can be used to display the competencies of a learner in the form of so-called competence states. The knots of this Hasse diagram indicate meaningful competence states of a student while the edges indicate admissible transitions from one competence state to another by acquiring another competency. In addition, the approach is based on a probabilistic view of having or lacking certain competencies. Very briefly, a Hasse diagram shows all possible (admissible) competence or knowledge states. The visualization in the form of Hasse diagrams, finally, allows identifying the learning paths, the history of learning, the present state, and – most importantly, to find proper recommendations for the next and the very next learning steps. In Year two we accomplished significant advancements of the Hasse diagram visualization feature for Learning Analytics and integrated them as an integral part of the Lea's Box system.

The Hasse diagram visualization tool is fully implemented and integral part of the system. In the final period, we completed the Hasse diagram visualization tool with the possibility to display the learning paths of individual students as well as the theoretical "average" path of an entire group. The details have been reported in deliverable D2.6.

4. Fast Learning Analytics for Superior Harnessing (FLASH)

During the final year, a relational database for a joint analysis of all data was constructed. Main goal is to provide a flexible and lasting infrastructure for gaining insight into data from various and multiple sources.

Example analyses have been performed for data

- from the platform adaptive curriculum (SEBIT)
- from the platform RAUNT (SEBIT)
- that had been pushed into the mylea_beta database by external tools and internally generated data from the portal

To achieve this goal, the database was planned as both relatively simple in structure and open-ended. Data to be pushed into the database should be easily identified regarding its origin, while remaining cleansed from all attributes that would prevent an analysis with data from other sources.

A full-fledged data warehouse, the best solution for achieving this, was deemed to costly in time, working hours and hardware to being finished during the lifetime of the project. A likewise approach was pursued in order to gain the maximum outcome for the resources at hand.

Regarding the original plan, the database was adapted during the final year as follows:

- additional information about students was made possible to be used as filter criteria or independent variables for analyses;
- groups were added as a dimension, as they showed to be important for users;
- schools or, more generally speaking, institutions/organisations were included, since they constituted a central orientation point for end-users, while being useful for research questions that are interesting for future use.

Summarised, the database services have been honed to firstly fit more snugly the structures that single end-users are familiar with from their daily practice. Secondly, additional layers of organisationally interesting dimensions have been added to expand the field of usage to administration.

The general structure and design idea of fast, simple and reliable performance has not been tampered with. School users need reliability over everything else, as the effective employment of their sparse time is crucial.

The basic layout of the data structure is:

- A student performs an action (learning/assessment) that is part of a curriculum in a given setting.
- It is as simple as this and as such closely maps the way in which teachers construct their own understanding of what is going on in the classroom or at club activities or even at home, during the work with an application.
- When the data comes with further information, this is taken into account. The database can make use of temporal information, which is seldom used in educational settings, as has been pointed out in the conference paper by Debus, Kickmeier-Rust, Albert (2016). Further capabilities that are part of the set of the database are analyses resting on competence-based knowledge structures, such as pre-requisites, thus enabling to map multiple pathways of learning unto a common goal.

An atomic skill or competency can be connected with various layers of super-skills, like concepts, topics, subjects or skill-groups, which themselves can be pushed into the database, should such configuration already exist for the data.

PROXY-OPENNESS

Rather than re-inventing the wheel, to make data directly pushable into or receivable by the FLASH database, a transfer from LEA'S BOX database was realised. The LB database had already established links to other tools, has been upgraded with an xAPI-standard entry point during the final year and was fully under control of the project consortium, which made development easy.

In this way, every tool that can connect with the LB's database, will have the capability to have its own data be analysed in FLASH. This helps to get the community started on FLASH, as it is not necessary to introduce it as a tool in its own right, but rather as an additional service layer of LB. As such, it strengthens the branding Lea's Box as the portal to provide in-depth analysis with a simple and easy-to-learn, fast-to-use design.

The inclusion of the xAPI-standard makes it possible to transmit data for years to come. This standard has gained wide acceptance during the final year. Being the follow-up standard of

SCORM its success is sure. This employment will strengthen the sustainability of the services funded by the EC and developed by the LB team.

The data is transferred according to industry standards.

1st Compare the relevant tables at point X in time

2nd The difference quantity is transferred to the FLASH database

3rd Should data get deleted in the LB database, it gets NOT deleted in the FLASH database – this has to be done separately

→ By this, we provide no backup-utility, although this could be a future implementation, but rather have an archive of students' performance ready.

4th Should errors occur in the FLASH database, then an error protocol is saved in a specific table in FLASH

Working Angles

Most learners' data storage system either employ a skill focus, what has been learned, or a learner focus, who is learning what.

To overcome this limitation, FLASH provides certain angles from which to analyse data:

1. Learner's data: track what an individual has done on a given point in time
2. Progress data: what has been learned when, without intense regard to a learner's property
3. Curriculum data: what is included and with which structure is it connected
4. Administrative data: overlook over classes, courses, teachers, subjects etc.

With these combined, the power of data warehousing, having a central storage that makes data available easily for intelligence applications from the classroom to the headmistress' office, it adds to the already impressive trailblazing capacity of Lea's Box to open the horizon for future computer-supported learning.

IMPLEMENTATION

The data analyses warehouse has been fully implemented as part of the Lea's Box system. Summarized, the idea of the entire data architecture and the data flow is that the main database (codename *myLEA*) is a flexible and comprehensive data store for general operations and analyses. For analyses requiring high computational performance (e.g., because large data sets are involved), the *FLASH* data warehouse is linked to the system. This data warehouse, mirrors data from the main database and pre-processes the data focussing on a set of pre-defined pedagogical questions (cf. D2.2). These analyses can be retrieved in a highly efficient way and in real time. myLEA and FLASH are connected through a synchronization gate that assures completeness and validity of data and that is avoiding duplications.

A set of predefined analytic queries have be implemented with the main database as well so that standard analyses can be called easily. The queries are listed in the annex.

5. TECHNICAL IMPLEMENTATION

Originally, the principle idea was to have the web platform (the box) that is equipped with an open interface to existing sources (i.e., tools, websites, apps, that are producing educationally relevant data). This data subsequently is processed within the platform and finally fed back to the user. The central component is a mechanism that controls the data flow and the deeper processing. In year one, we released the basic components (green; cf. Figure 11) and in year 2 we re-designed the system architecture and added further components (yellow). In year 3 we completed the envisaged developments and made the system mature and stable. Also new features have been added to the system (blue). The following figures shows the original architecture (as of DoW) and the extended system architecture as a result of the project.

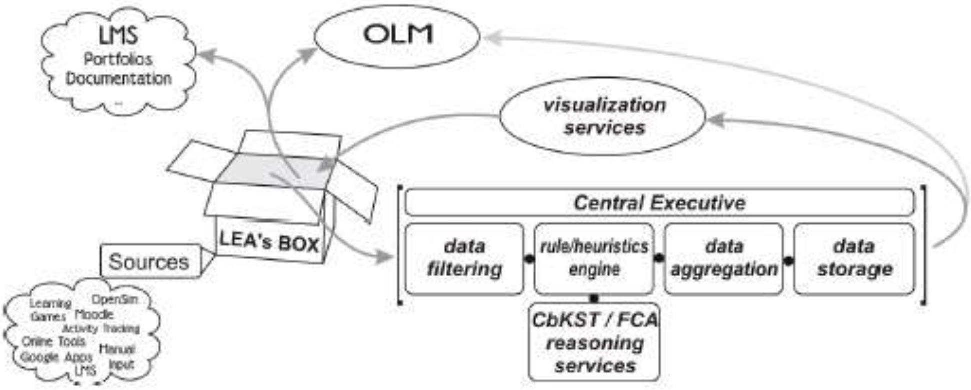

Figure 10. Originally proposed system architecture

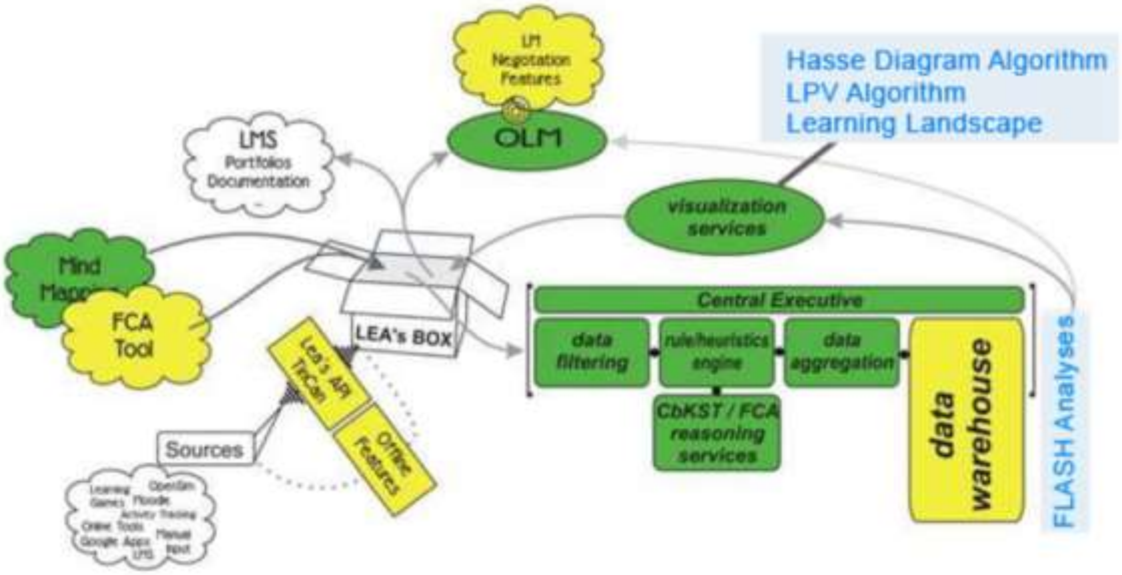

Figure 11. Final system architecture (green = Y1 yellow = Y2, blue = Y3)

REFERENCES

Albert, D., & Held, T. (1994). Establishing knowledge spaces by systematical problem construction. In D. Albert (Ed.), *Knowledge Structures* (pp. 78–112). New York: Springer Verlag.

Bedek, M., Kickmeier-Rust, M.D., & Albert, D. (2015). Formal Concept Analysis for Modelling Students in a Technology-enhanced Learning Setting. In: *Proceedings of the EC-TEL 2015 Workshop on Awareness and Reflection in Technology Enhanced Learning*, 27-33.

Bedek, M., Kopeinik, S., Pruenster, B., & Albert, D. (2015). Applying the Formal Concept Analysis to introduce guidance in an inquiry-based learning environment. In: *Proceedings of the 15th IEEE International Conference on Advanced Learning*, IEEE Press, New York, United States, 285-289.

Davey, B.A., & Priestley, H.A. (2002). *Introduction to Lattices and Order* (2nd edition). Cambridge: University Press.

Debus, K., Kickmeier-Rust, M.D., & Albert, D. (in press). Time Well Spent: Data Warehouse Solution for Temporal Analyses of Educational Data. To appear in the Proceedings of the International Conference on Computers in Education (ICCE) 2016.

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

Doignon, J.P., & Falmagne, J.C. (1999). *Knowledge Spaces.* Berlin: Springer.

Düntsch, I., & Gediga, G. (1996). On query Procedures to Build Knowledge Structures. *Journal of Mathematical Psychology*, 40, 160–168.

Heller, J., Steiner, C., Hockemeyer, C., & Albert, D. (2006). Competence-Based Knowledge Structures for Personalized Learning. *International Journal on E-Learning*, *5*, 75-88.

Heller, J., Ünlü, A., & Albert, D . (2013). Skills, competencies and knowledge structures. In: J.-C Falmagne, C. Doble, D. Albert, D. Eppstein, and X. Hu, (Eds), *Knowledge Spaces: Application in Education.* Berlin, Heidelberg: Springer.

Koppen, M. (1994). The construction of knowledge spaces by querying experts. In G. H. Fischer & D. Laming (Eds.), *Contributions to Mathematical Psychology, Psychometrics and Methodology* (pp. 137–147). New York: Springer–Verlag.

Korossy, K. (1997). Extending the Theory of Knowledge Spaces: A Competence–Performance Approach. *Zeitschrift für Psychologie*, 205, 53–82.

Korossy, K. (1999). Modeling Knowledge as Competence and Performance. In D. Albert & J. Lukas, (Eds.), *Knowledge Spaces: Theories, Empirical Research, and Applications* (pp. 103-132), Lawrence Erlbaum Associates, Mahwah, NJ.

Rusch, A., & Wille, R. (1996). Knowledge spaces and formal concept analysis. In H. H. Bock & W. Polasek (Eds.), *Data analysis and information systems: Statistical and conceptual approaches* (pp. 427-436). Berlin: Springer

Schrepp, M. (2003). A method for the analysis of hierarchical dependencies between items of a questionnaire. *Methods of Psychological Research*, *19*, 43-79.

Schrepp, M. (2006). ITA 2.0: A Program for Classical and Inductive Item Tree Analysis. *Journal of Statistical Software*, *16*(10), 1-14.

Spangenberg, N., & Wolff, K.E. (1991). Comparison between Biplot Analysis and Formal Concept Analysis of Repertory Grids, In H.H. Bock, H.H. & P. Ihm (Eds.), *Classification, Data Analysis, and Knowledge Organization* (pp. 104-112), Berlin: Springer.

Van Leeuwe, J.F.J. (1974). Item tree analysis. *Nederlands Tijdschrift voor de Psychologie*, *29*, 475-484

Wille, R. (1982). Restructuring lattice theory: An approach based on hierarchies of concepts. In I. Rival (Ed.), *Ordered sets* (pp. 445–470). Dordrecht: Reidel.

Wille, R. (2005). Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies, In B. Ganter , G. Stumme and R. Wille (Eds.), *Formal Concept Analysis* (pp. 1-34). Berlin: Springer.

ANNEXES

ANNEX I

Overview of mylea_beta transfer statements and pre-installed queries (deployed with the MySQL database)

	Query Name	***Query Result***
1	*subject_level_school*	levels that are used by each school and for each subject, to transfer as additional information for „group" dimension analyses
2	*student_levels*	in which levels students are at a given point, to be used as additional information for „people" dimension analyses
3	*people_teacher_subject*	which teachers have which subjects, to check validity of data (can this teacher be in this group or is data missing?)
4	*people_teacher_group*	which teachers are supervising which groupd, to be used as additional information for „group" dimension analyses
5	*people_studextid*	which students have external ids, to preserve them during transfer
6	*people_school*	which people are at which school at a given point (preparation of the set to transfer/update)
7	*n_user_school*	just for quantitative analysis: how many people are registered at which school
8	*n_teach_school*	just for quantitative analysis: how many teachers are registered at which school
9	*n_stud_school*	just for quantitative analysis: how many students are registered at which school

10	*n_admin_school*	just for quantitative analysis: how many administrators are registered at which school
11	*competency_subject_school*	which competencies are used in which subject by which school, complete with the competencies' parent competency (preparation of the set to transfer/update)
12	*competency_level_school*	which competencies are used at which levels by which school, additional information for „competency“ dimension analyses
13	*competency_group_school*	which competencies are used in which groups by which school, additional information for „competency“ dimension analyses, to be used as additional information for „group“ dimension analyses
	competency_dependentcompetency	dependent competencies for competencies, to completely transfer cbKST competency structures
	competency_datasource	datasources for competencies, prepared statement probably not necessary
	activity_school_datasource_subject	activity information, complete with school and datasource (preparation of the set to transfer/update)
	activity_competency_comp-school_activ-school	statement prepared for data integrity check
	IN_WORK_statements	under construction, code not ready for release

EXTERNAL ATTACHMENTS

- FCA-KST (Jürgen Heller, Michael Bedek, & Dietrich Albert)
- Knowledge spaces in the terminology of formal concept analysis (Reinhard Suck, Michael Bedek, & Dietrich Albert)