Draft of the article published as:

Vu T.M.H., Tchounikine P. (2021). Supporting teacher scripting with an ontological model of task-technique content knowledge. Computers & Education (163).

# Supporting Teacher Scripting with an Ontological Model of Task-Technique Content Knowledge


## Abstract

*Scripting is the phase of classroom management in which teachers build and/or fine-tune learning scenarios. When the learning objectives include task-technique knowledge, teachers face a specific difficulty: defining learning scenarios requires a holistic perspective of the different techniques, the types of tasks they address, and the interrelations between techniques and tasks, which may be highly complex for teachers. To address this issue, we developed a process for the semi-automatic elaboration of a task-technique knowledge model as an ontology, and tested it on a real scale case-study in mathematics education. We also designed interfaces that provide teachers with some access to this knowledge. A test shows that teachers found the proposal useful and usable.*


**Keywords:** Learning scenarios; Scripting; Task/technique knowledge; Ontology; Classroom orchestration.

## 1. Introduction

Scripting is the phase of classroom management in which teachers build and/or fine-tune learning scenarios, i.e., the tasks or exercises that will be proposed to learners. As emphasized by the orchestration perspective (Dillenbourg, 2013), scripting takes place before (primo-scripting) and, possibly, during the learning session (run-time scripting, i.e., on-the-fly modifications of the initial scenario according to learner achievement).

Scripting computer-based classroom activities is a core but challenging activity of teachers and, as such, must be supported (Persico, Pozzi, & Goodyear, 2018; Kollar & Fischer, 2013). This is one of the goals of the Learning Design field (Dagnino, Dimitriadis, Pozzi, Asensio-Pérez, & Rubia-Avi, 2018; Dalziel, Conole, Wills, Walker, Bennett, Dobozy, et al., 2016).

The state-of-the-art provides different supportive technologies for the representing/editing of learning scenarios (Celik & Magoulas, 2016; Dagnino et al., 2018; Pozzi, Asensio-Perez, Ceregini, Dagnino, Dimitriadis, & Earp, 2020). Many of these suggest a methodology to support “teachers as designers”, e.g., (Laurillard, Kennedy, Charlton, Wild, & Dimakopoulos, 2018). Other forms of scripting support include learning scenario patterns (see (Law, Li, Herrera, Chan, & Pong, 2017) or (Manathunga & Hernández-Leo, 2018) for recent examples) or automated design recommendations (Karga & Satratzemi, 2019). Support for the implementation of these scenarios is provided by specific platforms (Celik & Magoulas, 2016) and specific means to deploy them within different technical frameworks (see (Magnisalis & Demetriadis, 2017) or (Prieto, Asensio-Perez, Munoz-Cristóbal, Dimitriadis, Jorrín-Abellán, & Gomez-Sanchez, 2013)).

However, no attention has been paid to an aspect of scripting that may be highly complex for teachers, namely the need to reflect on task-technique content knowledge. A task is the work proposed to the learner, e.g., a mathematical exercise such as "solve $2x^2+3x=0$". A technique is a means for addressing this task, e.g., "factoring by inspection" or "completing the square". For a given set of techniques identified as learning objectives, the central aim of the scripting task is to identify the series of tasks that will make learners practice these techniques. This requires teachers to master task-technique knowledge, i.e., the different cases and sub-cases of tasks, which technique(s) may be used for a given task, which specific tasks can be achieved through the use of a given technique, and finally the different forms of concurrence among techniques.

Mathematics is a prototypical domain where task-technique knowledge is core (Kieran, 2019). In their highly influential paper, Ball, Thames & Phelps (2008) highlight how teachers require more than just the knowledge needed to solve the exercises to carry out their work. They also need the "mathematical knowledge for teaching" that is necessary to decide which task to start or continue with, anticipate how learners are likely to approach the task they assign to them, or analyze learner productions by answering questions such as "Is it okay to do this? Why? Would it work in general? Is it easier for some numbers and harder for others?" (Ball et al., 2008). Actually, these reflections are common to the preparation of exercises lists by teachers in many domains: learners practicing chemical analysis techniques are presented with the choice of techniques for the task of analyzing a chemical solution or a series thereof; learners practicing grammar or spelling rules have to complete a task of writing/correcting one or several sentences; etc.

In line with Ball et al's (2008) perspective, the overall objective of the work presented in this article is to promote and support good scripting practice, such as reflecting on the list of exercises (tasks) and how they may be achieved by learners (techniques). This includes supporting teachers' strategies such as reinforcing learners' knowledge by proposing tasks which are superficially different but may all be achieved with the same technique; motivating the need for a new technique by proposing a first series of tasks that are achievable with the techniques that learners already master then presenting a task for which the known techniques do not work, and which requires the new technique; or highlight that the technique a given learner is using is only suitable for the specific task he/she considers by adding a task it does not solve (run-time scripting).

The representation of such learning scenarios is satisfactorily addressed by state-of-the-art Learning Design languages. However, to elaborate or edit these scenarios, teachers also require a clear picture of the domain different techniques, how they interrelate, and how they relate to specific types of tasks – a highly complex undertaking when the domain includes a large set of techniques or types of tasks. There are still no means to support this aspect of scripting, which is an important issue. Studies in mathematics education consistently show that the teacher's mastering of such teaching knowledge contributes to gains in students' achievements; see for example (Hill, Rowan, & Ball, 2005). However, national and cross-national studies reveal an inadequate mastery of this knowledge in a significant number of teachers (see for example (Tatto, Rodriguez, & Reckase, 2020) or (Lui & Bonner, 2016)), which limits their capacities for instructional planning.

Supporting teachers in reflecting on task-technique knowledge requires a model of this knowledge. The state-of-the-art is to implement such a model as an ontology. The elaboration of such ontologies is a known bottleneck, which may be addressed following two approaches (Chang, D'Aniello, Gaeta, Orciuoli, Sampson, & Simonelli, 2020). One is to develop the model automatically via machine learning techniques, when this is possible. However, this approach produces models that cannot be processed by humans, which is prohibitive when the rationale is to offer a resource to teachers. The other is to develop the model with domain specialists, thereby ensuring that it is accessible by both humans and software components. This approach also guarantees coherence with the teaching setting. Typically, the mathematical knowledge taught in classrooms differs from one teaching institution to another; in order to be used by teachers, the knowledge model must be consistent with their setting. It is not therefore a question of building a universal knowledge model once and for all but rather developing the capacity to build and adjust models with the help of the appropriate domain specialists. The difficulty is the effort required for a "tailor-made" development of ontologies (Chang et al., 2020).

Through this analysis, two important research questions can be identified: (1) if and how one may provide support for education specialists during the elaboration of a domain ontology representing tasks, techniques and the interrelations between them, and (2) if and how one may provide teachers with usable interfaces to access this knowledge.

To answer these questions, we developed a process for the semi-automatic elaboration of a task-technique knowledge model as an ontology. The process builds from the Anthropological Theory of Didactics (ATD) model for addressing task-technique knowledge, but has a wider spectrum of application. We tested this process on a real scale case-study: supporting mathematics education specialists who were developing the task-technique knowledge model underlying the learning scenarios (in this case, series of mathematics exercises) for an existing tablet-based simulation. We then designed interfaces that provided teachers with some access to this knowledge, and conducted a usability test.

This methodology is coherent with some of the principles of Design Based Research (DBR; Anderson & Shattuck, 2012; Wang & Hannafin, 2005; Barab & Squire, 2004). A series of design actions are conducted to improve some educational practices. These actions are conceived not just to meet local needs, but to advance a theoretical agenda: understanding if and how improving teachers' capacity to manage "content knowledge for teaching" improves their scripting competence, and if and how ATD is a practical means for improving this capacity via the elaboration of task-technique knowledge models. The research questions studied in this article address intermediate milestones: is the elaboration of such models tractable and, given their complexity, can they be made usable by teachers? These two questions are addressed together because they form a coherent conceptual and practical step of the elaboration of such models, in this study and any other project requiring this type of model. In our case, the rationale for the work is not anchored in participants' (teachers') emerging demands, but rather in theoretical and empirical studies underlining that teachers must have a clear picture of task-technique knowledge. However, stakeholders are not simply treated as "subjects": the ontology elaboration process was co-conceived with mathematics education specialists, i.e., the actors involved in defining such knowledge models, and the design of the visualization interfaces was iteratively shaped with informant teachers, i.e., the final users. Coherence was ensured by recruiting these different actors from the same institution (regional education academy).

The rest of this article is organized as follows. Section 2 introduces the adopted knowledge representation and the running example. Section 3 describes the ontology construction process. Section 4 presents teacher interfaces. Section 5 presents an evaluation of these interfaces. Section 6 provides a general discussion. Section 7 presents the conclusions and some perspectives.

# 2. Theoretical background and running example

## *2.1. Theoretical background*

The Anthropological Theory of Didactics (ATD, Chevallard, 2007; Bosch, Chevallard, & Gascon, 2005) addresses knowledge as the science of a certain practice. It considers as the minimal unit of activity a "practice block" formed by a *task type* task and a *technique*, i.e., a method for achieving a task type. The task type notion captures the fact that a technique has a scope. For instance, "solve $2x^2+3x=0$" and "solve $2x^2+6x=0$" are two different tasks of the same task type: they may be solved by the same techniques. The fact that a given technique is part of the curricula is the rationale for presenting learners with tasks which may be solved by this technique.

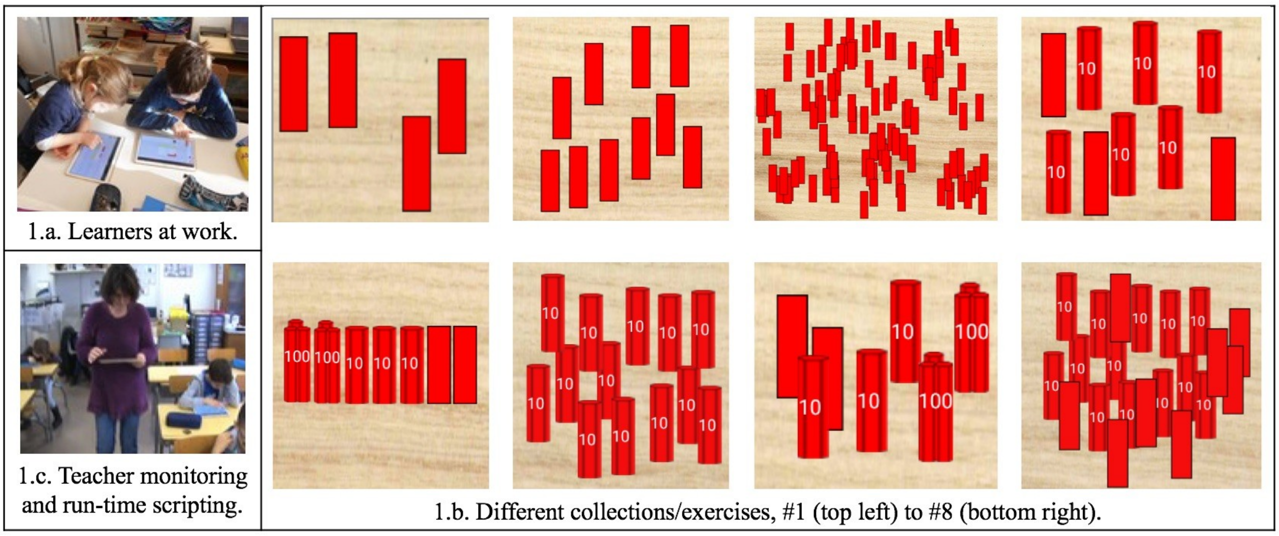
The ATD perspective is well in line with our research program. First, it focuses on the mathematical knowledge used in teaching practices, and acknowledges that such knowledge is relative to institutions. Second, its basic unit (type of task, technique) is perfectly suited to our interest (modeling task-technique knowledge). Third, a recent study has introduced the notion of variable (Chaachoua, Bessot, Romo, & Castela, 2019). Using this enhanced model, a task type is defined by an action verb and a complement taking its values in the considered domain notions. As examples: “Solve” + "a quadratic equation"; “Conjugate” + "a verb and a tense"; “Calculate” + "the voltage at the terminals of a dipole". As we will see in Section 3, characterizing this complement with variables permits the generation of the knowledge model by considering the different meaningful combinations.

We will now use our running example to illustrate the difficulty teachers experience when addressing a domain featuring different techniques, and how ATD may be used to model the knowledge at stake.

## *2.2. Running example: enumerating a collection with SimBuchette*

2.2.1 The SimBuchette simulation and the teaching setting

SimBuchette is a tablet-based simulation designed to train grade 1-3 learners in the competences required for the decimal number system (De Simone & Chaachoua, 2017). Our running example will be the use of this preexisting simulation for enumeration exercises (Tempier, 2016).



1.a. Learners at work.

1.c. Teacher monitoring and run-time scripting.

1.b. Different collections/exercises, #1 (top left) to #8 (bottom right).

**Figure 1.** The tablet-based simulation

In the considered setting, learning objectives include learners understanding the relations between hundreds, tens and units, and accurately using the different techniques that may be used to count the number of units in a collection. For this purpose, they are presented with a series of exercises on their tablets (Figure 1.a). An exercise corresponds to a specific collection, i.e., a particular number of sticks corresponding to units, tens or hundreds displayed on a virtual table (see different examples in Figure 1.b). The learner’s task is to enumerate the collection, i.e., calculate the overall number of units (10 in exercise#2, 138 in exercise#8). According to the exercises, this may involve basic techniques (e.g., counting one by one) or more complex ones, in particular when some conversions are needed. As a support, the simulation allows learners to bundle 10 units (or 10 tens) to obtain a ten (or a hundred); unbundle a ten (or a hundred) into 10 units (or 10 tens); or move the sticks around the table (e.g., to separate units and tens). All these actions are achieved via the tactile features of the tablet, which is well adapted to grade 1-3 learners.

Before the session, the teacher users her own tablet to define the list of exercises for each individual learner (primo-scripting). Such lists implement the kind of strategy we presented in Section 1, e.g., proposing an exercise such as exercise#2 (learners may count the sticks one by one) followed by a more complex such as exercise#3, where this technique does not work well. When the teacher launches the scenario, the lists of exercises are sent to the learners' tablets. The teacher monitors each learner activity and, if necessary/relevant, changes his/her list of exercises on the fly (run-time scripting; Figure 1.c).

2.2.2 Knowledge modeling

We will first present the modeling principles (as in this example a task is an exercise, we will simplify the reading of the text by using "exercise type" rather than "task type"). The considered exercises all correspond to the general "enumerate a collection" Exercise Type (ET for short). However, according to the characteristics of the collection, this ET can correspond to very different cases (see Figure 1.b). For instance, although exercises #1, #2 and #3 all solely involve units, they differ in nature. This is because they target different techniques. Typically, when the collection involves less than 5 units (as in exercise#1), it can be counted by subitizing, i.e., an immediate perception of the number. This is not the case for the 10 units in exercise#2, which learners enumerate by pointing to the sticks one after the other and counting from one to ten. Pointing and counting one by one, however, does not work well for exercise#3 (93 sticks, some of which overlap). This type of exercise calls for the use of more complex techniques. One of these is to group units in tens, count the latter, then add the remaining units. The counting of tens can be achieved using different techniques. One is to count the number of tens (e.g., 9) and convert them (9 tens make 90 units). Another is to count 10 by 10 (from ten to ninety).

The analysis of domain knowledge by mathematics education specialists led to the identification of no fewer than eighteen techniques (Jolivet, 2018). These eighteen different ways of enumerating a collection are interrelated in different ways. General techniques overlap with more specific ones. The scopes of certain techniques partially overlap. Finally, some techniques build on others. For instance, the counting of an unordered collection such as in exercise#7 may be achieved by ordering it (which leads to a collection similar to exercise#5) then applying the technique commonly used for an ordered collection. Sixteen of these techniques build on competences that correspond to learning objectives (e.g., different forms of conversion, knowing that 9 tens make 90 or counting 10 by 10). Two of them are odd techniques that some learners adopt, e.g., unbundling tens to count one by one.

The rationale for defining a specific ET is that it is likely to make learners practice a targeted technique (and/or realize that a technique one uses for some other exercises does not work well in this specific case). In other words, the ETs are defined according to the scope and/or pertinence of the identified techniques. This raises the issue of ET characterization. The mathematics education specialists identified four characterization items. Using the ATD extended model, each item is defined by a variable and its values (see Table 1). As a result, an ET is defined as "enumerate a collection" plus four values (one for each variable).

**Table1.** The variables.

| Variables | | Possible values (simplified: the values for each variables are not independent and, for some of them, hierarchically structured; there are 31 values in total) |
|---|---|---|
| v1 | Size of the collection | 0-5, 0-100, ... |
| v2 | Size is a multiple of 10 | yes/no |
| v3 | Bundles and spatial organization | bundles (yes/no), bundles made (all/partially), bundles and units are spatially ordered (yes/no), sticks overlap (yes/no), ... |
| v4 | Types of sticks | one type, several types |

The complexity revealed by this analysis well illustrates the two research questions described in Section 1. Although some of the technique or ET differences may appear subtle, it is crucial to take them into account in educational terms. Indeed, it is because these differences are subtle that teachers need support. The first issue is to build and represent a knowledge model featuring the different ETs, techniques, and interrelations. The high level of detail of the modeling creates a knowledge management issue: the four non-independent variables and the overall 31 values define a huge number of possible ETs. Although very few of them are of educational interest, they directly and indirectly relate one to the other in different ways and all of them must be identified to propose teachers a general picture. A purely manual elaboration of the model is subject to errors and omissions, if indeed it is possible. The second issue is to allow teachers to benefit from the knowledge model without being overloaded with information. One cannot expect teachers to invest an enormous amount of time in the understanding of a complex knowledge base. Its use must therefore be intuitive and straightforward.

In the next section, we show how such a domain model can be elaborated via a semi-automatized process and represented as an ontology. In Section 4 we will show how this ontology can be used to present teachers with relevant information via simple and intuitive interfaces.

## 3. The Ontology elaboration process

The process we propose is synthesized in Figure 2. It is a semi-automatic process, involving educationalists (in our case, mathematics education specialists) and algorithms. The output is an ontology which we implemented using the standard Protégé platform (Protégé, 2019). Protégé provides the means to describe knowledge models using a dedicated language named OWL and allows to reason on the represented data via a query system named SPARQL. Any other framework offering similar means may be used.

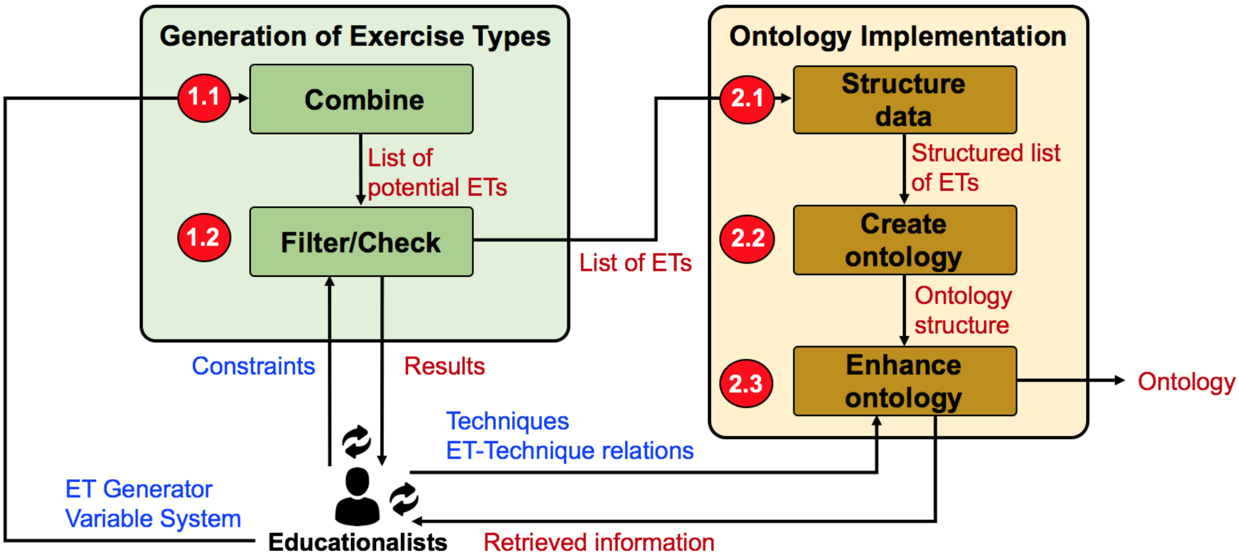



**Figure 2.** The ontology elaboration process

### *3.1. First phase: generation of the Exercise Types*

The first phase consists of identifying and structuring the ETs. The output is the list of coherent ETs.

The systematic combination of the different variables and their values allows the identification of all the potential ETs. However, only certain configurations make sense. In order to avoid meaningless combinations, two strategies are possible. The "smart strategy" is to elaborate a semantic model of how the different variables interplay and only generate meaningful configurations. The "brute force" strategy is to use a kind of cover-and-differentiate process (Eshelman, 1988) that generates all possible cases and then discards the meaningless ones. In our case, brute force appeared to be a good choice. The rationale is that elaborating a semantic model is a task educationalists are not necessarily familiar with. In direct contrast, reflecting on practical cases during the differentiation phase is closer to their competence, and much easier.

The implementation we adopted is as follows (see Figure 2, left-hand side). As a first step (1.1), all the different combinations are generated. Sticking to a simplicity objective, we used a spreadsheet application (Microsoft Excel). From an initial row in which the columns correspond to the different variables, a basic formula generates the different combinations of values as new rows. In this case, the 4 variables and their hierarchically organized values lead to an output of 2080 configurations/rows. The second step (1.2) consists of discarding the meaningless configurations. Here again, this may be implemented by formulas that apply to the different columns. As a basic example (translated, simplified): for collections formed with one type of sticks only, the notion of "ordered collection" (see the difference between #5 and #7 in Figure 1.b) is meaningless. These filtering formulas were co-elaborated by domain specialists expressing the semantic constraint and computer scientists translating it into a technical formula. As constraints can be applied individually, it is very easy to check which configurations are discarded and adapt the filter if it is too selective or wide-ranging. Overall, 16 constraints were created, reducing the 2080 potential configurations to 483 coherent ones. Many of the identified constraints retrospectively appeared to be evident. Nevertheless, this is only obvious *a posteriori*: it is much easier to identify odd configurations in a list than to create a semantic model in advance.

The output of this first phase is the list of all the consistent ETs. While a process based on educationalists' interviews may miss some configurations, this systematic process does not.

### *3.2. Second phase: the ontology implementation*

The second phase consists in the creation of the ontology. This ontology structure features the notions of the adopted model, i.e., Task/Exercise Type, Technique, and Variable (see Figure 3).

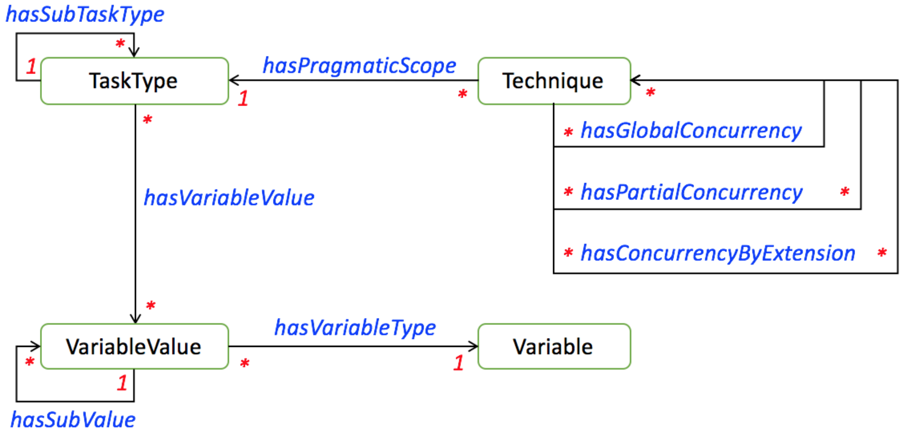



**Figure 3.** The ontology model (UML representation)

The process for creating the ontology is as follows (see Figure 2, right-hand side). The first step (2.1) consists of structuring the ETs in a hierarchical organization. The second step (2.2) consists of importing this structured data into Protégé (this is purely technical, and will not be detailed here). Finally, the third phase (2.3) consists in enhancing the ontology with the techniques.

Let us first explain the need for structuring the ETs (phase 2.1). The output of phase 1 is the list off all the consistent ETs (in our running example, 483 ETs). As already mentioned, most ETs are purely abstract cases: some of their sub-cases only correspond to specific techniques, and will be used to create exercises. As an example, the variable values corresponding to “collections of bundles that are already partially made” define an ET that may correspond to very different exercises. Teachers consider the important sub-cases associated to specific techniques only. Nevertheless, the abstract cases shed light on how ETs break into sub-ETs and thus indicate if and how the scope of their associated techniques overlap.

The modeling approach permits an automatized definition of the ETs hierarchical structure. A generic Java component takes the spreadsheet table containing all the consistent configurations as an input and uses variables analysis to identify the ET cases and sub-cases. The output is a hierarchy based on the generic ET "enumerate a collection" (no value attributed to any variable), and the branches correspond to sub- and sub-sub (etc.) cases. In our running example, this hierarchy has no less than 10 levels. This explains the difficulty educationalists and teachers have to grasp this complexity. As we will see in Section 4, smart interfaces can be used to retain these ETs in the knowledge base without overwhelming teachers.

Let us now study in detail the definition of ET-technique relations (phase 2.3). The knowledge model leads to consider four semantic relations.

Domain education specialists use the fundamental "hasPragmaticScope" relation to associate the techniques that constitute learning objectives with specific ETs (or, in other words, to indicate which ETs are likely to make learners practice the considered techniques). Defining technique $t_x$ as having the pragmatic scope to solve $ET_y$ means that when learners within the considered curriculum/grade and teaching organization are presented with $ET_y$, they are expected to use $t_x$ (or another technique with the same pragmatic scope). For instance, the pragmatic scope of the "count 1 by 1” technique corresponds to the ET illustrated by exercise#2 (Figure 1), and the pragmatic scope of both "count 10 by 10" and "count the number of tens and convert" corresponds to the ET illustrated by exercise#6 (Figure 1). Other techniques, typically of a more general nature, may also apply. However, these more general techniques are associated to other (more general) ETs, and when teachers want learners to use them, they present exercises of these general ETs rather than a specific sub-case. The domain education specialists define "hasPragmaticScope" relations from mathematics knowledge and "mathematical knowledge for teaching".

Three additional relations are needed to represent the different forms of concurrence among techniques stemming from the hierarchical structure of ETs (see Table 2). An ET is addressable by the technique(s) with which it is directly related via the "hasPragmaticScope" relation and the (more general) techniques associated to its more general cases (its ascendants in the ET hierarchical structure). Moreover, some of its sub-cases (i.e., some of its descendants) may be addressable by more specific techniques. Providing teachers with a clear picture of these technique concurrences is core for both scripting (e.g., to define an exercise that may be solved by technique $t_1$ and not by technique $t_2$) and monitoring (e.g., to detect that a learner has used a different technique to that expected and assess his/her process).

**Table 2.** Technique-ET and technique-technique relations.

| Relation | Interest | Examples |
|---|---|---|
| has Pragmatic Scope | Identify the ET(s) for which this technique is specifically suitable. | The Pragmatic Scope of the "subitizing" technique is the "less than 5 sticks" ET; the Pragmatic Scope of the "create bundles of 10 and count 10 by 10" technique is the "0-100 sticks, no bundles, multiple of 10" ET. |
| has Global Concurrency | Check which other techniques have the same scope, if any. | The "create bundles of 10 and count 10 by 10" and "create bundles of 10, count the number of bundles and convert" techniques are in Global Concurrency. |
| has Partial Concurrency | Check if any technique solves sub-cases of the case considered. | The “create bundles of 10 and count 10 by 10” and “create bundles of 10, count the number of bundles and convert” technique are in partial concurrency with the “count 1 by 1” (associated ET = “0-100 sticks, no bundles”) technique because they allow learners to solve the specific instances of this ET where the collection is a multiple of 10. |
| has Concurrency By_Extension | Check if a more general technique solves all instances of this case. | "subitizing" and “create bundles of 10 and count 10 by 10” techniques are in concurrency by extension with the "counting one by one" technique (but this general technique is not necessarily relevant/efficient for this cases). |

Although the concurrency relations are easy to understand in basic cases, studying how they apply to the different ETs and techniques is a difficult task (remember that the hierarchical structure has 10 levels). An ascendant, descendant or distant relation may link a pedagogically interesting ET to another interesting ET via several ETs with little pedagogical interest and, as several techniques may have the same pragmatic scope, relations may be multiple (the possible patterns are synthesized in Section 4, Figure 6). It would be difficult and even impossible to identify all the different relations by interviewing educationalists. A major interest of the ontology is to support this process by automatically calculating all the indirect relations via SPARQL queries.

The output of this second and final phase of the ontology elaboration is an operational representation of a teaching-oriented knowledge model of the domain. This model falls into the "task ontologies" category of Al-Yahya, George, & Alfaries' (2015) classification, i.e., ontologies relevant to a specific task (in this case, reflecting on the task type / technique relations).

# 4. Offering teachers easy access to the represented knowledge

In this section, we present interfaces that provide teachers with easy access to the knowledge represented in the ontology while preparing a learning scenario. The design of these interfaces stems from the ontology structure (i.e., the services that can be proposed through the theoretical knowledge model), from preexisting results related to the usability of table interfaces (Sobreira & Tchounikine, 2015; Prieto, Tchounikine, Asensio-Perez, Sobreira, & Dimitriadis, 2014), and from iterative improvements with informant teachers. The interface features two tables. The *consultation table* provides means to explore and reflect on the ETs, the techniques, and the relations between them. The *editing table* can be used to edit a scenario, i.e., select ETs, define precise exercises and attribute them to the different learners. Teachers may use the consulting table before beginning scenario editing and/or during this process if needed. To ensure the comprehension of the consultation table specific features, we first present the scenario editing table. The different interfaces are presented and explained in more detail in the "supplementary material" file associated to this article.

### *4.1. Editing a scenario*

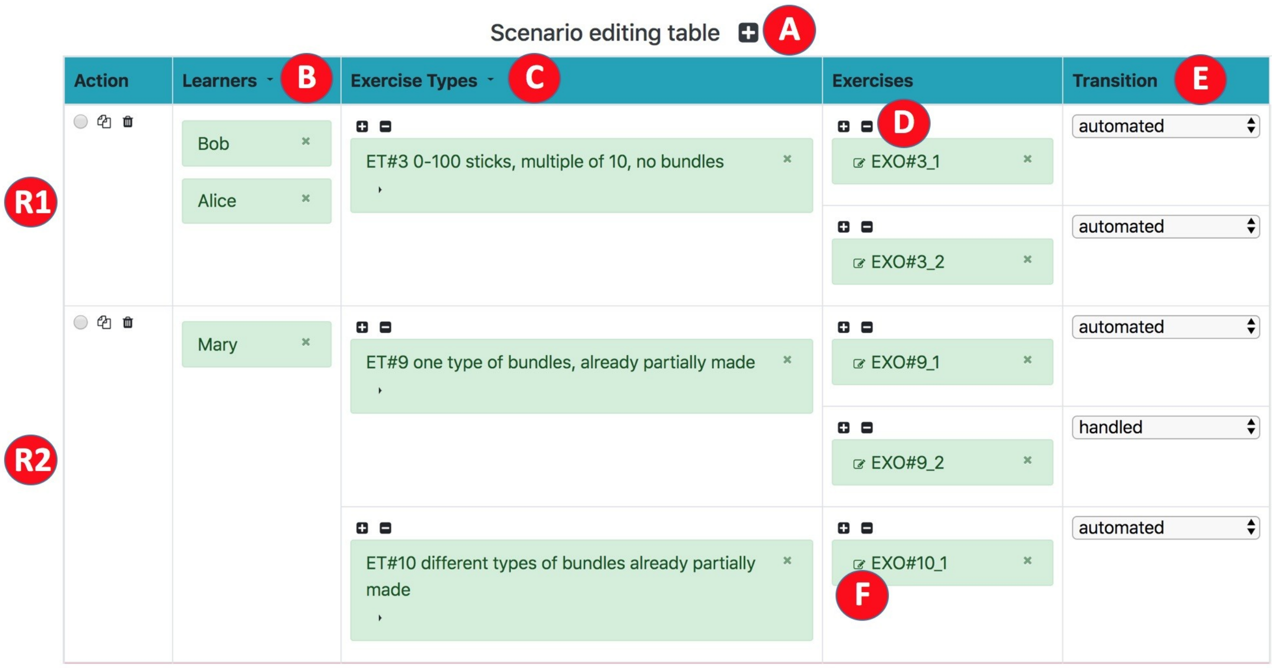


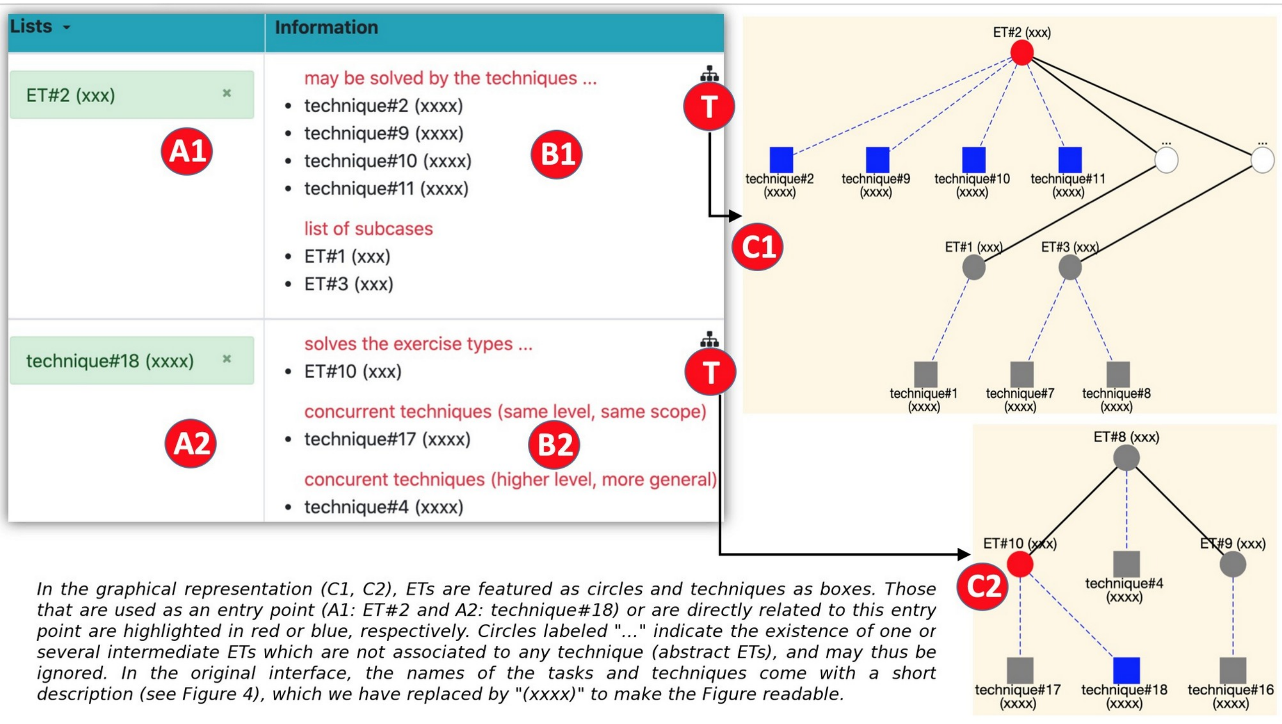

**Figure 4.** The scenario editing interface

The principle of the scenario editing table is that one row defines the tasks that will be presented to one group of learners. Following this structure, the editor permits the definition of a scenario as follows (see Figure 4): create a row ("+" button in **A**); select one or several learners in the drop-down list (**B**); select one or several ETs in the drop-down ET list (**C**); define the actual exercises chosen for each ET (**D**; see details in the following paragraph); state if the transition from one exercise to another is automated or handled by the teacher via his/her tablet (**E**). Figure 4 features an example with 2 rows (**R1** and **R2**). **R1** reads as: Alice and Bob will work on a single type of exercise ("ET#3 = exercises where the number of sticks is below 100 and is a multiple of 10, and no bundles have been made"; teachers can click on the ET names to see examples of these exercises, which in this case would open a pop-up similar to the exercise#3 in Figure 1); they have been attributed two exercises (EXO#3_1 and EXO#3_2), i.e., instances of ET#3 in which the number of sticks is set; the transition is automated, i.e., the system prompts the learners for the second exercise when they have finished the first. **R2** reads as: Mary will work on two successive types of exercises: ET#9 (two exercises, EXO#9_1, EXO#9_2) and then ET#10 (one exercise, EXO#10_1). The transition from the first to the second exercise is automatic, but the teacher will decide (via her tablet, cf. Figure 1.c) whether Mary will go directly to the third exercise.

When teachers select an ET (**C**), the fact that its characteristics are modeled in the ontology allows the automatic creation of exercises by generating a set of random values that respect the ET definition (number of sticks of each type, spatial organization, etc.). The teacher may add as many exercises as needed via the "+" button (**D**). In line with the philosophy that technology should be merely a support and that teachers have the final word on their teaching methods, the generated exercises may be edited (**F**). This opens a pop-up window that allows teachers to have a look at the random values and change them if needed. In order to support this process, two additional services are proposed. First, the exercise is visualized on a tablet as it will be presented to students, which helps teachers understand the configuration they have defined. Second, if any change is made, the system checks if the new configuration respects the ET constraints as defined in the ontology. Such practical features are good illustrations of how the design benefits from input from informant teachers and, in particular, their "in-context" analysis of how the ontology-based features can enhance their scripting needs. The table also offers different features (e.g., duplicating rows, copy-pasting exercises or customizing the textual description of the ETs and techniques) that facilitate script editing.

The simplicity of the editor allows teachers to focus on their core task, namely to decide which series of ETs and exercises they will attribute to the different learners. When teachers have a clear idea of which ETs they will include in the scenario, they simply select them from the drop-down list. When this not the case, they need to reflect on the ET definitions, their relations and the techniques they allow learners to practice. The consultation table, which is presented just above the editing table, is designed to support this process.

### *4.2. Reflecting on the ETs and techniques: the consultation table*




*In the graphical representation (C1, C2), ETs are featured as circles and techniques as boxes. Those that are used as an entry point (A1: ET#2 and A2: technique#18) or are directly related to this entry point are highlighted in red or blue, respectively. Circles labeled "..." indicate the existence of one or several intermediate ETs which are not associated to any technique (abstract ETs), and may thus be ignored. In the original interface, the names of the tasks and techniques come with a short description (see Figure 4), which we have replaced by "(xxxx)" to make the Figure readable.*

**Figure 5.** The consultation table

The consultation table is designed to provide teachers with a simple and intuitive interface featuring the ETs, the techniques, and their relations. The look and feel is similar to the scripting interface (see Figure 5 and supplementary material). When teachers select an ET (e.g., **A1**), the system features a prototypical case as a vignette (see Figure 1.b), the variables values and an overview of the direct and indirect relations between the ET concerned and other ETs and techniques (**B1**). When teachers select a technique (e.g., **A2**), the system features a textual description, the ETs it allows to address (basic usage of the hasPragmaticScope relation) and an overview of the direct and indirect relations between the technique considered and other ETs and techniques (**B2**). The possibility to create several rows allows teachers to easily browse different ETs or techniques, which is an important feature for supporting holistic reflections.

The description of the ET-techniques relations is offered via two modes: as a text (**B1**, **B2**), and as a graphical structure (**C1**, **C2**) when the learner clicks on the tree icon (**T**). This graphical structure features the relevant neighborhood of the ET or technique considered only (the complete 10-level hierarchical structure is far too large to be visualized and would not be useful for the teachers anyway).

Figure 6 presents as abstract patterns all the different possible organizations of ETs and techniques that may occur. Our running example actually included one or several cases of pattern#1 to pattern#8 cases. This once again illustrates that despite the apparent simplicity of some domains, the knowledge concerned may reveal unexpected complexity for which teachers need support.

Let us now illustrate how such information may support the scripting strategies described in Section 1. Pattern #3 suggests and supports analyses such as "*As the objective is to make learners practice t, one should avoid exercises corresponding to sub-cases such as ET#1 or ET#2*" (these may be addressed by specific techniques); "*In order to highlight the scope of general technique t, one may first include one exercise per sub-case (ET#1, ET#2, ...), monitor/scaffold learner activity to ensure they master the corresponding specific techniques, and then highlight how the general technique t covers all cases*"; "*In order to highlight the scope of the general technique t, one may first include exercises that can be solved by the specific techniques, and then introduce exercises for which these techniques do not work*". Pattern #5 suggests and supports analyses such as: "*As ET#1 is a sub-case of ET, it may be relevant to introduce ET#1 exercises and then exercises for the other ET sub-cases*", or "*A strategy for making learners master t1-1 may be, in addition to presenting ET#1 exercises that it can solve, to introduce exercises of sub-cases (e.g., ET#2) that it can not solve*".

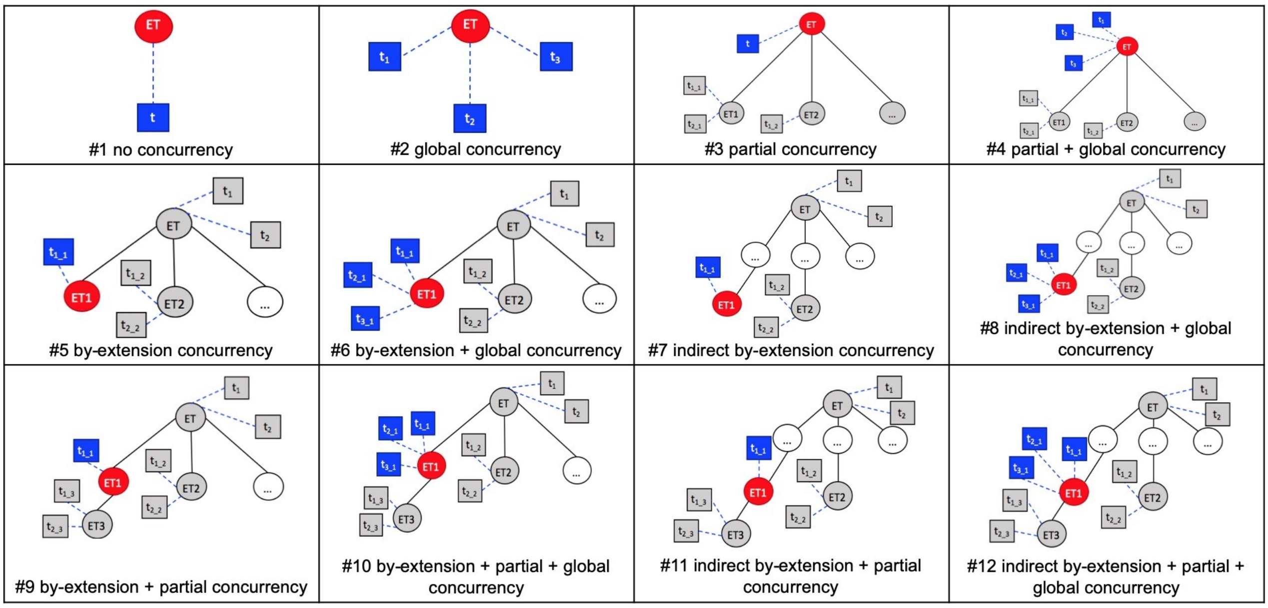



**Figure 6.** Direct and indirect relations between ETs and techniques

## 5. Evaluation

A test was set up to evaluate the consultation table and the script editor usability. The panel included "standard" primary schools teachers as targeted in this work, i.e., in-service teachers who are not engaged in any research project, and who have no specific experience with ontologies or research-based learning design tools. They were recruited from different schools. All participants had experience in teaching the "enumerating collections" topic. To avoid any issue related to the tablet simulation, we first reviewed the simulation features and the exercises it allows with the tested teacher. The teacher was then presented with a normalized presentation of the consultation table and the script editor, and invited to use the system. S/he first freely explored the system, and was then asked to use it in three prototypical cases: "If you want to check how ET#9 exercises may be solved, how would you proceed? How do you interpret the input provided by the system?"; "If you want to define a scenario for making learners practice technique#4, how would you proceed?"; "If you want to edit a script where Learner#1 is offered one ET#9 exercise and Learner#2 and #3 are offered two ET#10 exercises and then one ET#11 exercise, how would you proceed?). The teacher then completed a questionnaire that addressed usability (5-point Likert scale, from unusable to very easy to use) and perceived usefulness (5-point Likert scale, from useless to very useful). Finally, s/he was engaged in an open debriefing discussion. Although a sample of 5 testers is considered sufficient in usability tests where answers are coherent (Virzi, 1992), which was the case in our study, we tested 8 teachers to confirm data saturation (5 teachers were tested individually, then 3 additional teachers completed the questionnaire individually before a collective debriefing discussion).

All panel members spontaneously highlighted their general interest in being supported in their scripting activities. Several mentioned that the system rationale made them realize the importance of anticipating learners' possible solving strategies, and their lack of "mathematical knowledge for teaching" to do so. All of them were impressed by, and highly interested in, the level of detail of the techniques and ET analysis. They showed particular interest in the fact that apparently equivalent ETs actually called for different techniques, and thus differed in nature. Several spontaneously expressed a desire to benefit from such analyses for many of the topics they teach, including non-mathematical disciplines.

The panel rated use of the consultation table featuring the ETs and techniques as easy (a quarter of the panel) or very easy to use (three quarters of the panel), and all participants rated it as very useful. Interestingly, different behaviors were observed. We expected teachers to use the table to first select techniques (which are the learning objectives), consider the ETs that were appropriate to make learners practice these techniques, then finally consider the different direct and indirect relations between ETs and techniques (e.g., overlapping scopes and concurrences). However, some teachers mainly used ETs as the entry point. Open discussion revealed that while these teachers admitted that explicit (rather than implicit) consideration of the techniques as the learning objectives was good practice, it was not their current practice.

The panel rated the utility of the different ET - technique relations as summarized in Table 3. The results are globally coherent, but it is interesting to note that the utility of the relation among ETs is judged "somewhat useful" by some teachers and "very useful" by some others. With respect to how this information is presented, all teachers agree that a textual presentation is easy or (predominantly) very easy to use. The results are more balanced for the tree structure (3 teachers ranked it as "somewhat easy" to use only, and the others as easy or very easy to use).

The script editor was rated as easy (3 answers) or very easy (5 answers) to use. This confirms previous empirical studies (Sobreira & Tchounikine, 2015; Prieto, Tchounikine, Asensio-Perez, Sobreira, & Dimitriadis, 2014).

Finally, the capacity of the system to generate exercises from the ET specification was perceived by teachers as an important added value. Here again, several spontaneously expressed a desire to benefit from such features in many of the topics they teach.

**Table 3.** Perceived usefulness of ET - technique relation (responses of the 8 teachers).

| | useless | poorly useful | somewhat useful | useful | very useful |
|---|---|---|---|---|---|
| Link ET / possible techniques allowing the learner to solve it | | | | 2 | 6 |
| Link ET / more general and more specific ETs | | | 3 | 2 | 3 |
| Link technique / the ETs it allows learners to solve | | | | 2 | 6 |
| Link technique / concurrent techniques | | | 1 | 3 | 4 |

# 6. Discussion

## *6.1. The ontology and its elaboration*

The added value of the semi-automated process we presented is (1) to make the elaboration of the ontology tractable, and (2) to limit the risk of elaborating a shallow, incomplete or possibly incorrect expertise model. It is not possible to empirically compare the time needed with and without the semi-automated process support, if only because the latter is prohibitively time consuming. An unsupported process would require several times the couple of dozen hours that were necessary for the example we presented. Moreover, given the number of combinations and the complexity of the interrelations, it is likely to fail and/or involve errors or omissions.

The approach we presented specifically considers task/technique knowledge. It may be combined with other more general forms of support for ontology development, see for example (Wang, Brendan, & Ogata, 2017) or (Chang et al., 2020). As an example, applying text-mining to textbooks may help identify learning objectives, recurrent exercises and/or some characterizing variables.

The automatic specification of exercises is a research topic in its own right, and this study does not address it as such. Nevertheless, it can be noted that an important finding in this research field is that automatic generators are most useful when the generated exercises are of predictable difficulty (Chen, Zilles, West, & Bretl, 2019). An ontology such as the one presented in this article is a suitable resource to address this need.

## *6.2. Teacher interfaces*

The added value of the designed interfaces is to offer teachers a simple and dedicated access to the specific data they need. In direct contrast with non-specific interfaces (e.g., the Protégé interface), they avoid overwhelming teachers with unsuitable information (e.g., abstract ETs), technical aspects (e.g., the notion of SPARQL query) and inappropriate/meaningless features (such interfaces are far too complex for non-specialists). Moreover, the data is provided in a way which is directly related to the teachers' need (elaborating scenarios), interoperated with the scenario editor, and in line with their work practice. The evaluation shows that they found the proposal useful and usable.

This result shows that when seeking to help teachers reflect on task-technique knowledge, it makes sense to elaborate an ontological model of the required knowledge that is based on a theoretical background (ATD) and a detailed didactic analysis, then provide access to this knowledge via interface structures which are known to be easily usable by teachers (in our case, table interfaces). Nevertheless, iteratively enhancing the interfaces and the services offered with the help of informant practitioners was a key step of the design process. It enabled us to design tools that were in line with the design culture of the teachers and the institution, a key criterion (Dagnino et al., 2018; Laurillard, Charlton, Craft, Dimakopoulos, Ljubojevic, Magoulas, Masterman, Pujadas, Whitley, & Whittlestone, 2013; Demetriadis, Barbas, Molohides, Palaigeorgiou, Psillos, Vlahavas, Tsoukalas, & Pombortsis, 2003). Moreover, this approach allows identifying additional services with high practical value for teachers (e.g., exercises generation), making the overall proposal attractive.

The fact some teachers focused on techniques and some others on ETs is aligned with other works that have revealed how teachers may engage in scripting from different starting points, e.g., (Albó & Hernández-Leo, 2018). From a design perspective, this provides a rationale for supporting both entry points. From a scientific perspective, this calls for further investigations to understand which factors may play a role.

Another interesting point is the somewhat balanced evaluation of the visual (tree) presentation of the ET-technique relations. The debriefing sessions revealed that some teachers "read" the tree representation very easily, but others have difficulties with this format. In direct contrast, none of them had any difficulty with the textual presentation. This finding is coherent with the literature review presented in (Dagnino et al., 2018), which recalls that simplicity and coherence with teachers' practice are of core importance. Although tree or graph representations they are easily read by Computer Scientists, who use them in their basic practices, this is not necessarily the case for all teachers.

### *6.3. Scope and limitations*

The work presented in this article applies to domains for which a task-technique model is relevant (i.e., domains in which tasks may be addressed by different techniques) and useful (this is an important concern when designing learning scenarios). Task-technique modeling is used in the ATD theoretical approach, which originated in mathematics education but has since been used in other domains (Ladage, Achiam, & Marandino, 2019). As examples, Tetchueng, Garlatti, & Laube (2008) used ATD to model physics know-how, and Girault & d'Ham (2014) used it in chemistry to model the design of experimental procedures. Actually, modeling knowledge in terms of tasks and means (referred to as techniques or methods) is a rather general approach (Choquet, Danna, Tchounikine, & Trichet, 1998), and our proposals are not therefore limited to the use of the ATD theory.

The ontology notions (task-type, variables, techniques and the different types of relations) are abstract. They may thus be used across different domains. This is also the case of the relations featured in the ontology (and in the teacher interface), which derive from the ET-technique relations and the notion of pragmatic scope. The table representation we used is just one option.

An intrinsic limitation of the proposed ontology elaboration approach is the necessary involvement of domain educationalists. Although some teachers may of course have the competence to build ontologies, it cannot be expected from all of them. As soon as one goes into detail, the task becomes demanding. Actually, this is the rationale behind our work. If the knowledge at stake is basic there is no need to create an ontology, because the teachers do not require specific support.

One limitation of the consultation table usability test is that the experimental setting was elaborated with a focus on usefulness. In order to present teachers with the innovative ontology-based services and check if they perceived them as useful, we used a structure (the table interface) that we expected to be usable. We did however check this usability to control any potential negative impact on the usefulness test. The fact that the teachers did indeed consider the table interface to be usable confirms that it is one possible design option. Nevertheless, defining the best interface would require the evaluation of different designs (Tohidi, Buxton, Baecker, & Sellen, 2006). This type of analysis could lead to the identification and availability of a range of options for teachers, in a similar way to how the ETs-techniques relations were offered via both a textual and a graphical representation. Care must also be taken to ensure that the interfaces offer all the additional services that teachers could find useful. When considering orchestration settings, another limitation is that the consultation table tests were conducted for the primo-scripting use-case only, i.e., preparing a learning scenario *before* the session. This is explained by our experimental setting. Previous works showed that the table interface allows teachers to manage on-the-fly changes/inclusions of exercises (Wang, Tchounikine, & Quignard, 2018). However, learners in our experimental setting spend only a few minutes on each exercise. This creates time pressure and, as a consequence, it is unlikely that teachers have time to consider the consultation table during the teaching session: changing or adding an exercise at run-time is an opportunistic contextual decision based on their pre-session analysis. Run-time usage of the consultation table should be studied in another type of setting in which it makes practical sense for teachers.

## 7. Conclusion and perspectives

In domains including task-technique knowledge, scripting requires teachers to have a holistic perspective of the different techniques, how they interrelate, and how they relate to specific types of tasks. This may prove to be highly complex and require support. We presented an example illustrating this complexity, the necessity of elaborating a knowledge model and the difficulty of this task. We raised the research question of if and how education specialists can be supported in the elaboration of such a model as an ontology, and showed that this may be addressed via a semi-automated process. We also raised the research question of if and how teachers could be provided with usable interfaces to make use of this knowledge, and showed that this can be addressed via smart interfaces exploiting the ontology semantic direct and indirect relations. The main takeaways of this article are: (1) the identification and illustration of how the teaching strategies used for task-technique knowledge lead to a specific and difficult scripting issue; (2) the importance of addressing this issue; (3) the means to elaborate task-technique knowledge models; and (4) the means to provide an easy access to these models. The adopted knowledge model and the design principles may be used across different domains. The ontology elaboration process requires the involvement of domain specialists. This limitation may be mitigated by the existence of preexisting analyses (typically, if a detailed curriculum exists) and/or additional means such as text-mining techniques.

These results pave the way for considering if and how the use of these knowledge models can help teachers to define more relevant scripts and attain better learning outcomes. The input provided by teachers during the usability test shows they are willing to benefit from such supports, which resonates with studies of teachers' acceptance/resistance attitudes considering the infusion of technology into schools: in addition to enhancing the quality of their teaching, an important high order goal for teachers is competence in their job (Demetriadis et al., 2003). Nevertheless, the quality of the scripts that teachers produce via this support is another question, and requires a specific study.

A related but different perspective is to study if/how the ontology and/or the patterns (and examples of analyses) we featured in Figure 6 may be used in teacher training. As providing a means to share with colleagues is an efficient way to facilitate adoption of learning technologies (Laurilard et al., 2018), another perspective is to study if/how teachers may be supported in sharing the rationale for the scenarios they elaborate (rather than simply the scenarios).

Taking a more general perspective, we believe that teachers' activities in relation with the use of learning technologies must be addressed as a chain: any difficulty with a task may hinder engagement. Learning scenario editors proved their value to support teachers in elaborating and representing scenarios (Dagnino et al., 2018; Dalziel et al., 2016; Demetriadis et al., 2003). However, in some domains, teachers also need to reflect on the content knowledge. In such cases, the "support for reflection" issue must be studied as such, and not simply from the scenario representation perspective. We studied this subject for the specific and arguably complex case of task-technique knowledge, and call for works that develop other approaches to this scripting issue. This includes studying how it may be articulated with monitoring. Typically, future studies should explore if/how an ontology such as that studied in this article may also be used as a resource for automated diagnosis of learners' behaviors.

## 8. References


Albó, L. Hernández-Leo, D. (2018). Identifying design principles for learning design tools: the case of edCrumble. Proceedings of the 13th European Conference on Technology Enhanced Learning, EC-TEL 2018, Leeds, UK.

Al-Yahya, M., George, R., & Alfaries, A. (2015). Ontologies in E-learning: Review of the literature. International Journal of Software Engineering and its Applications 9(2), 67–84.

Anderson, T., & Shattuck, J. (2012). Design-based research: A decade of progress in education research?. Educational researcher, 41(1), 16-25.

Ball, D. L., Thames, M.H., & Phelps, G. (2008). Content Knowledge for teaching: What makes it special? Journal of Teacher Education, 59(5), 389–407.

Barab, S., & Squire, K. (2004). Design-based research: Putting a stake in the ground. The journal of the learning sciences, 13(1), 1-14.

Bosch, M., Chevallard, Y. & Gascon, J. (2005). Science or magic? The use of models in didactics of mathematics. In M. Bosch (ed.), Proceedings of CERME4, 4th Conference of European Research in Mathematics Education, pp. 1254-1263. Sint Feliu de Guixols, Spain.

Celik, D., & Magoulas, G. D. (2016). A review, timeline, and categorization of learning design tools. In: D.K.W. Chiu, I. Marenzi, U. Nanni and M. Spaniol, M. (eds.) Advances in Web-Based Learning – ICWL 2016. Lecture Notes in Computer Science 10013. New York, U.S.: Springer, pp. 3-13.

Chaachoua, H., Bessot, A., Romo, A. & Castela, C. (2019) Developments and functionalities in the praxeological model. In M. Bosch, Y. Chevallard, F. Javier Garcia and J. Monaghan (eds) Working with the anthropological theory of the didactic: A comprehensive casebook, pp. 41-60. London: Routledge.

Chang, M., D'Aniello, G., Gaeta, M., Orciuoli, F., Sampson, D., & Simonelli, C. (2020). Building Ontology-Driven Tutoring Models for Intelligent Tutoring Systems Using Data Mining. IEEE Access, 8, 48151-48162.

Chen, B., Zilles, C., West, M., & Bretl, T. (2019). Effect of discrete and continuous parameter variation on difficulty in automatic item generation. *In International Conference on Artificial Intelligence in Education (pp.* 71-83). Springer, Cham.

Chevallard, Y. (2007). Readjusting didactics to a changing epistemology. European Educational Research Journal 6, 9-27.

Choquet, C., Danna, F., Tchounikine, P., Trichet, F. (1998). Modelling Knowledge-Based Components of a Learning Environment within the Task/Method Paradigm. In: Lecture Notes in Computer Science (Intelligent Tutoring Systems), ITS'98, 1998, San Antonio (USA), p. 56-65.

Dagnino, F. M., Dimitriadis, Y., Pozzi, F., Asensio-Pérez, J. I., & Rubia-Avi, B. (2018). Exploring teachers' needs and the existing barriers to the adoption of learning design methods and tools: A literature survey. British Journal of Educational Technology 49, 998–1013.

Dalziel, J., Conole, G., Wills, S., Walker, S., Bennett, S., Dobozy, E., et al. (2016). The Larnaca declaration on learning design. Journal of Interactive Media in Education, 1–24.

Demetriadis, S., Barbas, A., Molohides, A., Palaigeorgiou, G., Psillos, D., Vlahavas, I., Tsoukalas, I. & Pombortsis, A. (2003). "Cultures in negotiation": teachers' acceptance/resistance attitudes considering the infusion of technology into schools. Computers & Education 41(1), 19-37.

De Simone, M., & Chaachoua, H. (2017). The transposition of counting situations in a virtual environment. In G. Aldon & J. Trgalova (eds) Proceedings of the 13th International Conference on Technology in Mathematics Teaching, pp. 314-322. Lyon, France.

Dillenbourg, P. (2013). Design for classroom orchestration. Computers & Education 69, 485-492.

Eshelman, L. (1988). MOLE: A knowledge-acquisition tool for cover-and-differentiate systems. In: Automating knowledge acquisition for expert systems, pp. 37-80. Springer, Boston, MA.

Girault I., & d'Ham, C. (2014). Scaffolding a Complex Task of Experimental Design in Chemistry with a Computer Environment. Journal of Science Education and Technology 23 (4), 514-526.

Hill, H. C., Rowan, B., & Ball, D. L. (2005). Effects of teachers' mathematical knowledge for teaching on student achievement. American educational research journal, 42(2), 371-406.

Jolivet, S. (2018). Modèle de description didactique, de ressources d'apprentissage en mathématiques, pour l'indexation et des services EIAH. PhD thesis (in French). Univ. Grenoble Alpes, France.

Karga, S., & Satratzemi, M. (2019). Evaluating Teachers' Perceptions of Learning Design Recommender Systems. In European Conference on Technology Enhanced Learning, pp. 98-111. Springer, Cham.

Kieran C. (2019). Task Design Frameworks in Mathematics Education Research: An Example of a Domain-Specific Frame for Algebra Learning with Technological Tools. In: Kaiser G., Presmeg N. (eds) Compendium for Early Career Researchers in Mathematics Education. ICME-13 Monographs. Springer, Cham.

Kollar, I. & Fischer, F. (2013). Orchestration is nothing without conducting – But arranging ties the two together!: A response to Dillenbourg. Computers & Education 69, 507-509.

Ladage, C., Achiam, M. & Marandino, M. (2019). Research on ATD outside mathematics. In M. Bosch, Y. Chevallard., F. Javier Garcia, and J. Monaghan (eds) Working with the anthropological theory of the didactic: A comprehensive casebook, pp. 12-30. London: Routledge.

Laurillard, D., Kennedy, E., Charlton, P., Wild, J., & Dimakopoulos, D. (2018). Using technology to develop teachers as designers of TEL: Evaluating the learning designer. British Journal of Educational Technology, 49(6), 1044-1058.

Laurillard, D., Charlton, P., Craft, B., Dimakopoulos, D., Ljubojevic, D., Magoulas, G., Masterman E., Pujadas, R., Whitley, E.A. & Whittlestone, K. (2013). A constructionist learning environment for teachers to model learning designs. Journal of computer assisted learning 29(1), 15-30.

Law, N. W. Y., Li, L., Herrera, L. F., Chan, A., & Pong, T. C. (2017). A pattern language based learning design studio for an analytics informed inter-professional design community. Interaction Design and Architecture (s).

Lui, A. M., & Bonner, S. M. (2016). Preservice and inservice teachers' knowledge, beliefs, and instructional planning in primary school mathematics. Teaching and Teacher Education, 56, 1-13.

Magnisalis, I., & Demetriadis, S. (2017). An Architecture Combining IMS-LD and Web Services for Flexible Data-Transfer in CSCL. IEEE Transactions on Learning Technologies 10(2), 205-218.

Manathunga, K., & Hernández-Leo, D. (2018). Authoring and enactment of mobile pyramid-based collaborative learning activities. British Journal of Educational Technology, 49(2), 262-275.

Persico, D., Pozzi, F., & Goodyear, P. (2018). Teachers as designers of TEL interventions – Editorial of special issue. British Journal of Educational Technology, 49, 975–980.

Pozzi, F., Asensio-Perez, J. I., Ceregini, A., Dagnino, F. M., Dimitriadis, Y. & Earp, J. (2020): Supporting and representing Learning Design with digital tools: in between guidance and flexibility. Technology, Pedagogy and Education (in press).

Prieto, L. P., Asensio-Perez, J. I., Munoz-Cristóbal, J. A., Dimitriadis, Y. A., Jorrín-Abellán, I. M., & Gomez-Sanchez, E. (2013). Enabling teachers to deploy CSCL designs across distributed learning environments. IEEE Transactions on Learning Technologies, 6(4), 324-336.

Prieto, L. P., Tchounikine, P., Asensio-Perez, J. I., Sobreira, P., & Dimitriadis, Y. (2014). Exploring teachers' perceptions on different CSCL script editing tools. Computers & Education, 78, 383–396.

Protégé. http://protege.stanford.edu (last retrieved Dec. 2019).

Sobreira, P., & Tchounikine, P. (2015). Table-Based Representations Can be Used to Offer Easy-to-use, Flexible, and Adaptable Learning Scenario Editors. Computers & Education, 80, 15–27.

Tatto, M. T., Rodriguez, M., & Reckase, M. (2020). Early career mathematics teachers: Concepts, methods, and strategies for comparative international research. Teaching and Teacher Education, 103118.

Tempier, F. (2016). New perspectives for didactical engineering: an example for the development of a number resource for teaching decimal number system. Journal of Mathematics Teacher Education 19(2), 261-276.

Tetchueng, J. L., Garlatti, S., & Laube, S. (2008). A Context-Aware Learning System based on generic scenarios and the theory in didactic anthropology of knowledge. International Journal of Computer & Applications 5(1), 71-87.

Tohidi, M., Buxton, W., Baecker, R., & Sellen, A. (2006). Getting the right design and the design right. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06). Association for Computing Machinery, New York, NY, USA, 1243–1252.

Virzi, R. A. (1992). Refining the test phase of usability evaluation: How many subjects is enough?. Human factors, 34(4), 457-468.

Wang, F., & Hannafin, M. J. (2005). Design-based research and technology-enhanced learning environments. Educational technology research and development, 53(4), 5-23.

Wang, J., Brendan, F., & Ogata, H. (2017). Semi-automatic construction of ontology based on data mining technique. In: International Conference on Learning Technologies and Learning Environments, pp. 511-515.

Wang P., Tchounikine P., Quignard M. (2018). Chao: a framework for the development of orchestration technologies for Technology-Enhanced Learning activities using tablets in classrooms. International Journal of Technology Enhanced Learning 10 (1/2): 1-21.