Using Semantic Waves to Analyse the Effectiveness of Unplugged Computing Activities

Paul Curzon
p.curzon@qmul.ac.uk
Queen Mary University of London
London, UK

Jane Waite
j.l.waite@qmul.ac.uk
Queen Mary University of London
London, UK

Karl Maton
karl.maton@sydney.edu.au
The University of Sydney
Sydney, Australia

James Donohue
j.donohue@mmu.ac.uk
Manchester Metropolitan University
Manchester, UK

ABSTRACT

We apply the notion of ‘semantic waves’ from Legitimation Code Theory (LCT), a powerful educational framework, to Computer Science Education. We consider two case studies exploring how a simple analysis can help improve learning activities. The case studies focus on unplugged activities used in the context of both teaching school students and teacher continuing professional development. We used a simple method based on LCT to analyse the activities in terms of their ‘semantic profiles’: changes in the context-dependence and complexity of the knowledge being taught. This led to improvements to the activities. We argue that ‘semantic waves’, or moves back and forth between concrete/simpler and abstract/complex knowledge, help show ways that an unplugged activity might be effective or not, and how small changes to the activities can make a difference in potentially offering a more fruitful learning experience.

CCS CONCEPTS

• **Social and professional topics → K-12 education**; **CS1**; **Adult education**.

KEYWORDS

education, unplugged, semantic waves, teacher professional development

ACM Reference Format:
Paul Curzon, Jane Waite, Karl Maton, and James Donohue. 2020. Using Semantic Waves to Analyse the Effectiveness of Unplugged Computing Activities. In *WiPSCE ’20: The 15th Workshop in Primary and Secondary Computing Education, October 28–30, 2010, Online.* ACM, New York, NY, USA, 10 pages.

1 INTRODUCTION

Computer Science is being introduced at school level worldwide, but with little existing research into appropriate pedagogy, and with many teachers having little experience to build on. Different teaching approaches have emerged with varying degrees of success. Simple ways are needed to help teachers to predict the effectiveness of lesson plans and identify ways to improve them.

One pedagogical approach is ‘unplugged computing’, where physical, tangible activities or demonstrations are used to explain abstract, intangible computing concepts as well as allowing students to explore powerful ideas. This approach is now popular, though evidence of its effectiveness is mixed.

The contribution of the paper is to show how Legitimation Code Theory (‘LCT’) [21, 22] can be used to review and improve computer science learning activities. We apply a heuristic version of an analytical method called ‘semantic profiling’ from LCT. It involves drawing heuristic figures of the profiles of activities. We applied it to two computer science case studies involving unplugged demonstration-style activities. The method aims to be simple enough for teachers to apply in their pedagogy; to that end, it re-contextualises a more detailed analytical procedure, concentrating on the teacher’s approach in the lesson rather than the learners’ experience. We provide a series of simple but key LCT questions for teachers to focus on when analysing activities. We show how this quick and simple analysis can suggest where changes could improve an unplugged computer science lesson plan and how it can help reflection on a teaching session for future improvement. We finally summarise a set of ‘packing’ and ‘unpacking’ examples which highlight the technical nature of academic knowledge in the context of computer science.

In the remainder of the paper, we first give an overview of LCT, including its application to computing. Second, we describe the specific method we applied to analyse lesson plans. Third, we discuss two detailed case studies in turn. For each, we describe the activity and initial lesson plan, the LCT analysis, resulting improvements to the delivered lesson, and further reflection from analysis conducted after lessons were delivered. Finally, we summarise the kinds of packing and unpacking of academic knowledge encountered in the case studies.

For published version see ACM online WiPSCE ’20, October 28–30, 2020, Online

ACM ISBN xx…$xx

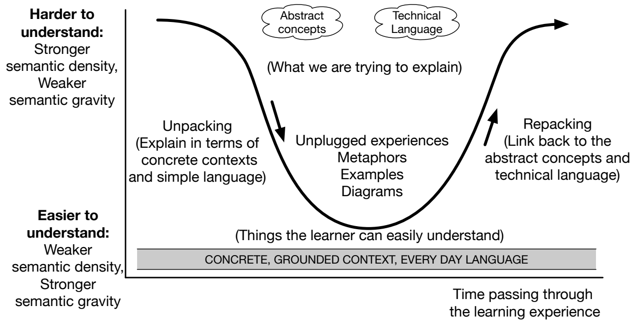

Figure 1: A semantic profile showing a semantic wave

2 LEGITIMATION CODE THEORY

Legitimation Code Theory ('LCT') is a sociological framework for analysing social practices [21, 22]. It has been successfully applied in many disciplines to both research and support teaching and learning. Semantic profiling, which forms part of LCT, focuses on changes in the context-dependence of knowledge ('semantic gravity') and the complexity of knowledge ('semantic density'). It offers a valuable means of systematically reflecting on what makes an effective learning experience.

Semantic profiling is where strengths in the semantic gravity and semantic density of the knowledge being expressed in a classroom are traced over time. An explanation easily understandable by a 'novice' is likely to involve concrete, context-dependent ideas (stronger semantic gravity) and everyday language expressing non-technical knowledge (weaker semantic density). In contrast, explanations understandable to experts involve abstract, context-independent concepts (weaker semantic gravity) and highly technical, complex knowledge (stronger semantic density). The key to mastery of a subject is the ability to move between concrete, simpler knowledge and more abstract, complex knowledge. This is what LCT terms 'semantic waves' (see Figure 1).

These ideas can be used to provide a 'semantic profile' of a specific learning experience, such as in Figure 1, showing strengths of semantic gravity and semantic density on the y-axis and time passing on the x-axis. The profile shows changes in the context-dependence and complexity of the knowledge being expressed. This can be used to plot an actual lesson plan or lesson as delivered. Different shapes of semantic profiles suggest different learning experiences. By drawing coarse-grained semantic profiles, focusing on the steps in the lesson plan, and just the direction of change rather than quantitatively determined levels, learning experiences from different lessons can be reflected upon, discussed and compared, and improvements suggested.

Studies using LCT show that learning experiences that involve semantic waves, or moves between these forms of knowledge, are a valuable way of enabling a learner to build their mastery of a subject [23]. One kind of semantic wave is first to introduce the abstract language and technical concepts to be covered, 'unpack' these concepts through giving concrete contexts and simpler language, and then 'repack' the ideas by linking back to the abstract concepts and technical language to be mastered (see Figure 1). One way of unpacking is to connect abstract ideas to concrete examples and break down complex knowledge into component ideas, often expressed in everyday language. Repacking is where the concrete, simpler knowledge is then linked back to more abstract and technical knowledge.

Studies using these ideas also suggest some potentially poor semantic profiles [23]:

1. **High flatlining**: the learning experience involves only abstract concepts and technical language with dense meanings. Explanations on the web by experts (e.g., Wikipedia or stack-exchange style explanations) can have this flavour. There is little unpacking or repacking of knowledge.
2. **Low flatlining**: the learning experience involves only concrete, simpler knowledge, often expressed in everyday language, with no explicit links to the abstract, technical knowledge of the learning outcomes. There is little unpacking or repacking of knowledge.
3. **Down escalators**: Abstract, technical concepts are unpacked into more concrete, simpler knowledge but not linked back up after the activity. There is no repacking. Explanations by new Computer Science undergraduates often have this profile.

3 LITERATURE REVIEW

Computing lesson plans have been analysed with respect to the content covered (e.g. [6]) and detailed lesson plans are used in general as the constraining factor when comparing interventions, particularly in lesson studies [17]. Research indicates that when developing lesson plans, teachers do not always consider how their learners will think [7].

LCT gives a way to reflect on why different teaching approaches do or do not work. It can be used to evaluate individual, or sequences of, lesson plans and online resources. It can also be used to teach students how to write good explanations. It has been applied across disciplines from Chemistry to Ballet [5, 16]. The utility of using semantic profiles in understanding the teaching of Computing has been argued for [10, 11, 14, 15, 34]. However, it has previously only been applied to a small number of computing activities. [34] showed you could plot the semantic profile of a successful unplugged activity, and that it had a wave structure. [18] explored the use of LCT, and particularly semantic waves, when investigating how teachers integrate ICT and educational technology into their teaching practice. More recently, [19] is exploring the knowledge practices of teaching coding using LCT as a lens. We expand here on a previous presentation [11], giving detail to an example introduced there as well as a new case study. We thus provide further evidence of LCT's applicability to computing contexts.

Unplugged computing teaches concepts without using a computer [2]. Approaches include using magic tricks, puzzles, role play and storytelling [12, 13]. By using objects and physical activities to represent complex concepts, the intangible is made tangible, and ideas that are normally only described verbally can be pointed at and manipulated. Physical enactment entrenches memories helping to link the remembered experience to the abstract concept [1]. As well as a physical aspect, unplugged activities often use analogies and metaphors. These draw upon familiar contexts to situate the teaching of complex concepts. However, the question of what is the most effective pedagogy to teach computing remains open [33] and despite unplugged computing being popular, especially in the context of primary and secondary (K-12) school education [29] there is mixed evidence as to the effectiveness of this approach [27, 32].

We adopt a case study method in this paper. The case study method is recommended for its versatility in the detailed description and analysis of activities [25, 31] and so case studies are suited for the in-depth review and improvement of lesson planning using simplified profiling. This case study approach has been used in LCT analysis of the teaching of university biology to review an individual lesson before and after revision to improve the teaching [26].

4 METHOD

To analyse an activity outline or lesson plan as a semantic profile, we examined the activity step-by-step for changes in semantic gravity and semantic density, drawing a heuristic version of the profile. Whether there were changes up or down or not were determined in a broad-brushed and heuristic manner. In simple terms, we asked whether the step in the presentation or activity involved concrete and simpler meanings (often expressed in everyday language) or more abstract and complex meanings (often expressed in technical language). Where technical words are used, the profile is higher than where only everyday language is used; similarly, an explanation that remains abstract is higher than one with concrete or tangible examples. Our focus was thus on creating a relative profile that revealed shifts between the forms of knowledge being expressed rather than plotting absolute levels of context-dependence and complexity. It should be emphasised that the analysis does NOT focus at all on the amounts of the shifts up or down at each point, and the resulting plot is just a sketch in this sense. This increases its utility for inexperienced analysts such as teachers. Whilst there are means of being able to analyse the strengths of semantic gravity and semantic density down to the word level in great detail [24], this degree of detail is unnecessary, indeed counter-productive, here, given the pedagogic purpose.

Having drawn it, we asked three key questions of the profile:

QUESTION 1: Does the shape of the profile plotted follow a rough wave shape (either 'u'– or 'n'–shaped), avoiding 'flatline' and 'down escalator' profiles, in order to support learners to move between concrete, simpler knowledge and more abstract, complex knowledge? (It is worth noting that waves are not necessarily smooth and perfectly 'u'– or 'n'–shaped).

QUESTION 2: How far up and down does the semantic profile move? This is to consider whether the teaching is encouraging learners to fully engage with the complex, technical knowledge (high) and fully connect with simpler explanations and concrete examples (low).

QUESTION 3: Who is doing the packing and/or unpacking that moves knowledge up and down the profile: the teacher or the learner (or both)? This considers whether teachers are modelling these shifts or learners are also being encouraged to themselves engage with technical knowledge and concrete examples.

We applied this semantic wave analysis method to two activities: "Teleporting robots" [9] and "Box Variables" [8], developed by the first author, and that he has delivered many times over many years in different contexts. He used semantic waves to reflect on each before, and then after, delivering them as activities within new continuing professional development (CPD) talks for teachers. For each, the profiles were plotted in the simple, heuristic way described. The three questions were then applied to help improve the initial lesson plan immediately prior to delivery. Afterwards, the questions were again applied to aid reflection on further improvements.

When thinking about the profile of a learning activity, we must keep in mind how individual learners will respond to the intended experience. Students will bring with them differing prior learning, knowledge and experience, levels of self-efficacy and degree of intrinsic motivation [28]. Here our learners were motivated teachers with classroom experience as well as a wealth of life experience. Our focus, however, is on the teaching rather than the experience of individual learners.

5 CASE STUDY 1: THE TELEPORTING ROBOT

5.1 The original lesson plan

The first case study was part of an online talk serving as CPD for teachers that included a complete sample computing activity called 'The Teleporting Robot Magic trick' [9]. This was presented as an activity that the teachers could do themselves with their classes. It involves a trick based around a 6-piece jigsaw of a picture of 17 robots when initially built. Mix the pieces and put them back in a different way, however, and there are now only 16 robots, despite the pieces being the same. It is a self-working trick: anyone can do it just by following the instructions even if they do not know how it works. The activity aims to explain, in an introductory way, the nature of an algorithm, using the magic trick as a route into explaining concepts concerning algorithms.

The original plan involved the following steps.

(T1) Set out the learning objective of being able to explain an algorithm.
(T2) Introduce that we were going to use a magic trick to help do this.
(T3) Do the trick.
(T4) Explain the steps followed to do the trick (but not why it works), pointing out that the learners can follow them to do the trick themselves without knowing how it works.
(T5) Explain how self-working tricks in magic are algorithms and so like programs.
(T6) Summarise the meaning of algorithm.

5.2 Analysis: Initial profile

Analysing the semantic profile of the planned activity, and answering the three key questions, led to changes to the version actually presented.

QUESTION 1: Is it a wave? The profile of the teleporting robot activity does follow a wave structure as shown in Figure 2 (Note the curves drawn, following the method, are heuristic and no absolute values of quantitative changes should be inferred from them). However, there is no detailed unpacking activity planned, only a single explicit link step, Point (T2) in Figure 2, between the learning

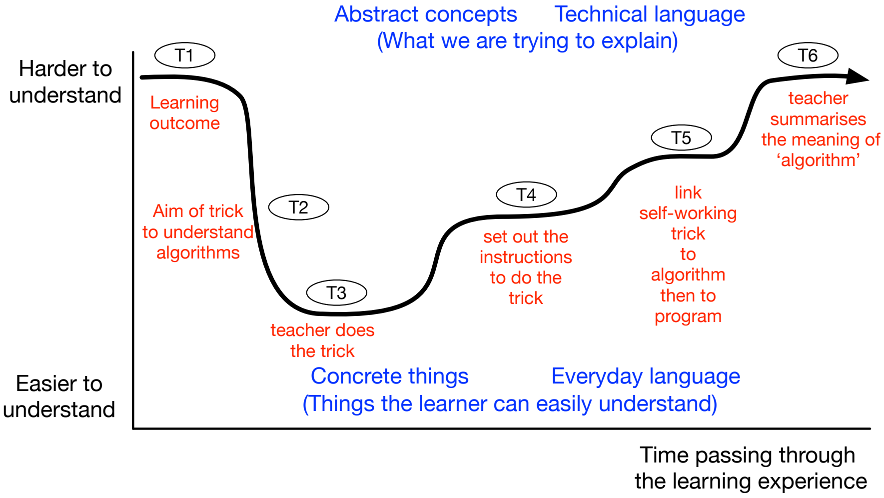

Figure 2: The semantic profile of the initial plan of the Teleporting Robot activity. Numbers, (T1), (T2), etc refer to steps in the original plan and are used in the explanatory text.

outcome (T1) and the trick being performed (T3). This was intentional as the approach followed the TEMI [20] idea of setting up a mystery (the trick) that is then explored. It was felt that, therefore, going straight to the trick was important so as not to reveal in advance what was to be discovered. This lack of unpacking puts more emphasis on the need for strong repacking later.

Repacking occurs in a series of steps. First, repacking involves explaining that the trick is a sequence of steps (T4), moving away from performance and introducing the idea of following steps precisely. Slightly more technical computing language is introduced but set within the example of a magician's self-working trick. Second, the activity shifts higher (T5) by linking this to the technical word 'algorithm' and linking what audience members would be doing when undertaking the trick to the technical context of what a computer is doing when following instructions. Third, the summary (T6) shifts higher still by further explaining an algorithm in technical and abstract terms.

QUESTION 2: How high and low do you go? The trick itself is a concrete, tangible example, a physical performance accompanied by non-computing language. The profile thus reaches low. Conversely, the final explanation returns to the original learning outcome and explains the abstract concepts using technical language, reaching relatively high.

QUESTION 3: Who is packing? The unpacking and repacking is largely driven by the teacher rather than by learners.

5.3 Improvements

After considering the three questions above, the plan was changed prior to presentation (see Figure 3). Changes aimed to ensure the activity involved more repacking. Activities were added (T4a, T5a, T5b, T6a, see Figure 3) to help learners engage with shifting up from simpler, concrete knowledge towards more complex, abstract knowledge. Learners were given access to the magic jigsaw (either physically or online) and encouraged to try the trick themselves (T4a) before further packing was done. This involved them doing some initial packing in turning the presentation of the trick into instructions they personally would then follow. This was also intended to ensure that the point made later: "you could do the trick even though you have no idea how it works" is embedded in and shifting upwards from their own experience. In addition, the single original slide linking the self-working trick to an algorithm and to a program (T5) was split into two new distinct slides that created two repacking steps. The first (T5a) focused on repacking via switching to explaining the trick using more technical computer science language, suggesting the trick is a form of algorithm. The second (T5b) step moved from talking about tricks to talking about programs in the abstract and suggesting that computers also must follow algorithms. Finally, rather than just summarising at the end (T6), an extra step was added before (T6a) that asks learners to write their own summary explanation of an algorithm with an example.

5.4 Further Reflection

Figure 3 portrays changes in the knowledge to be expressed in the planned presentation. However, the actual presentation looked different. By accident, the presenter did not show a slide that both introduced the learning outcome of algorithms (T1) and explained how it would be demonstrated with a trick (T2). This slide was intended to link understanding algorithms to the trick. The presenter noticed his mistake and backtracked, realising because of the semantic profile this would have implications for how the knowledge expressed would be sequenced. Had the presenter not realised so not corrected the error, the activity would have followed the profile shown in Figure 4. It would have started at the bottom of a curve with an introduction to the trick (T1a) and shifted only upwards.

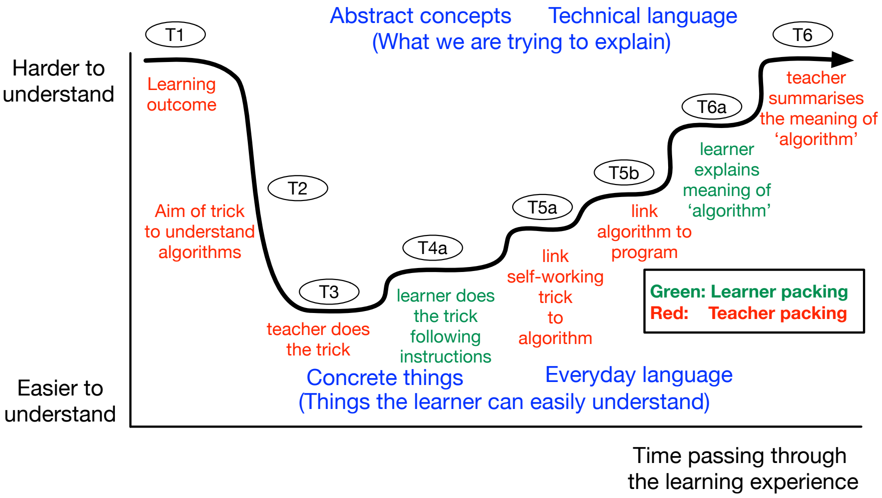

Figure 3: The semantic profile of the plan of the Teleporting Robot activity as presented.

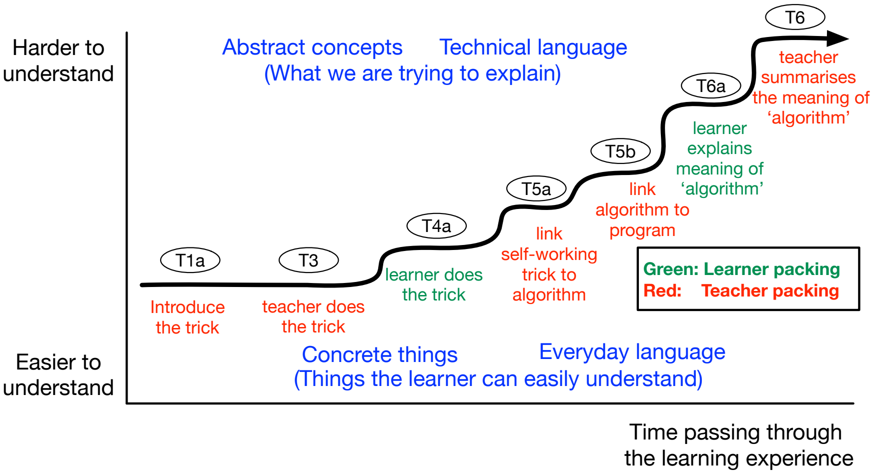

Figure 4: The semantic profile of the presentation of the Teleporting Robot activity as it was nearly mistakenly presented without the introduction.

This has the potential for making the audience wonder why they were watching a trick and also potentially be less prepared to do the repacking that leads back up to algorithms. It is likely this would not have provided quite so successful a lesson for all learners.

The error led the presenter to reflect further about unpacking and its importance and to re-evaluate whether some explicit unpacking could be introduced without ruining the mystery and so the point of the trick. For example, an initial simplified overview could have been given introducing what an algorithm is in terms of it being represented through a set of instructions that if blindly and precisely followed guarantee a desired effect. This could help prepare learners, when later following the steps of the trick, to work out for themselves that the trick they were doing was something akin to an algorithm. This could also mean they could do more repacking themselves before the teacher did so for them. The unpacking would therefore scaffold the later repacking, supporting learners' entry into the more technical, complex knowledge of computing. This new version of the activity thus becomes:

- **(T1)** Set out the learning object of being able to explain an algorithm.
- **(T2)** Introduce that we are going to use a magic trick to help do this.

(T2a) Outline in simple language that an algorithm is "a set of steps…"
(T3) Do the trick.
(T4a) Explain the steps followed to do the trick (but not why it works) and have the learners do the trick, pointing out that they can do it themselves without knowing how it works.
(T5a) Explain how self-working tricks in magic are algorithms.
(T5b) Explain that programs implement algorithms in a programming language for a computer.
(T6a) Have the learner summarise the meaning of 'algorithm'.
(T6) The teacher summarises the meaning of algorithm.

This has the semantic profile of Figure 5.

5.5 Summary of the case study

The advantages of the final version over the original, in terms of offering a potentially more fruitful learning activity, are that:

- There is now staged unpacking helping the learners to later link self-working tricks with algorithms themselves.
- There are more repacking steps that make the repacking more gradual and in clearer stages.
- The steps engage the learner to themselves do the repacking of experiences into concepts, rather than passively listening to the teacher attempting to do so.

6 CASE STUDY 2: BOX VARIABLES

6.1 The original lesson plan

"Box Variables" [8], the second unplugged programming activity analysed, was chosen as it was known to be highly effective (from student feedback and peer review).

The learning outcome of the activity is an understanding of the concept of variables and the precise effect of executing sequences of assignments in the context of introductory programming in Python. It uses as a concrete, technical example the swapping of values between two variables. The activity involves role play, with learners acting as variables, based on a metaphor that a variable is a box with integrated shredder and copier. Values are copied and ripped to shreds at appropriate points in the role play.

Variations of the activity have been used for many years (for example teaching undergraduates introductory programming), there is a video of the activity for wider use in school contexts, and it has been presented repeatedly as part of teacher CPD sessions demonstrating it as an activity they can do when teaching programming. We analysed a version of the planned activity, to be delivered in a face-to-face context as part of a physical teacher CPD workshop. This led to changes to that delivery.

The initial lesson plan was:

(B1) Set out the learning outcome.
(B2) Explain the variable as a metaphorical box with integrated shredder and copier.
(B3) Introduce an example ('swap') program.
(B4) Introduce the role play acting out the execution of the program.
(B5) Explain each line of code in turn as it is acted out.
(B6) Summarise lessons about the meaning of variables and assignment and their execution.

6.2 Analysis: Initial profile

QUESTION 1: Is it a wave? The profile of the planned Box Variables activity (see Figure 6) exhibited a series of semantic waves within a larger wave.

After the learning outcome is stated (B1) there is an initial unpacking step (B2) where the meaning of the word variable is first described in terms of the everyday context of the box. This is an unpacking step, but the later concrete example of using it to role play stepping through a program descends further still. It is followed by a repacking step (B3) where it is linked to a concrete program using highly technical programming language. However, only limited links are made at this stage. Direct links are made to the concrete example one line at a time in a series of waves in the subsequent role play (B5), before finally summarising the meanings of the words 'variable' and 'assignment' including the effect of executing sequences of assignments in a program (B6).

This sequence of shifts in the context-dependence and complexity of the knowledge being expressed are intended to support the packing and unpacking of each separate sub-concept corresponding to the lines of code. Studies in LCT suggest this to be a strong approach for introducing learners into the complex constellations of meaning that comprise academic knowledge because it allows each meaning to be explained separately and then brought together to give a coherent picture of the whole [23].

QUESTION 2: How high and low do you go? The overview at the end links all the way back to the original learning outcome and the abstract, complex concepts of variables and assignments (B6). This is the outer or overall wave traced by the profile. The start and the end thus reach relatively high, into the complex and abstract knowledge of computing. Conversely, the role play (the mini waves) takes learners to everyday and concrete notions of practically putting things in boxes, copying them and shredding them. This is made even more tangible in the role play by acting out these practices (the bottom of the waves at B5). As a whole, the profile does reach relatively high and low. However, after B5 the activity shifts abruptly upwards, reflecting a rather speedy repacking to reach the top of the profile.

QUESTION 3: Who is packing? The unpacking and repacking is largely driven by the teacher rather than by learners. Three learners are directly involved in the role play and thus directly involved in unpacking and repacking concepts by doing actions with boxes that correspond to the program line being considered. However, other learners are not actively engaged. Thus, there is limited opportunity for encouraging learners to engage with shifts in knowledge.

6.3 Improvements

The analysis highlighted that the final, overall repacking section was weak. Whilst the analysis method itself does not suggest specific repacking methods, further reflection suggested asking learners to individually summarise the points learned (B6a) and so explicitly repack what they have seen. This led to a new lesson plan used as the basis of the presentation (Figure 7):

(B1) Set out the learning outcome.
(B2) Explain the variable as a metaphorical box with integrated shredder and copier.

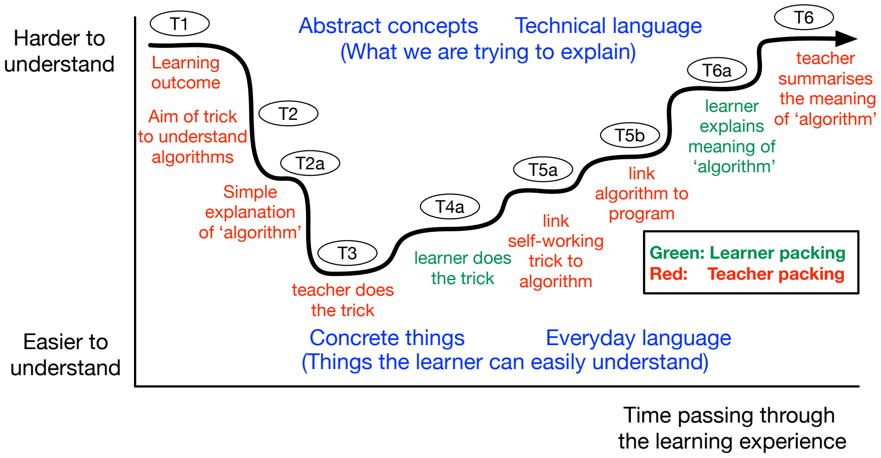

Figure 5: The semantic profile of the final plan of the Teleporting Robot activity for future presentation.

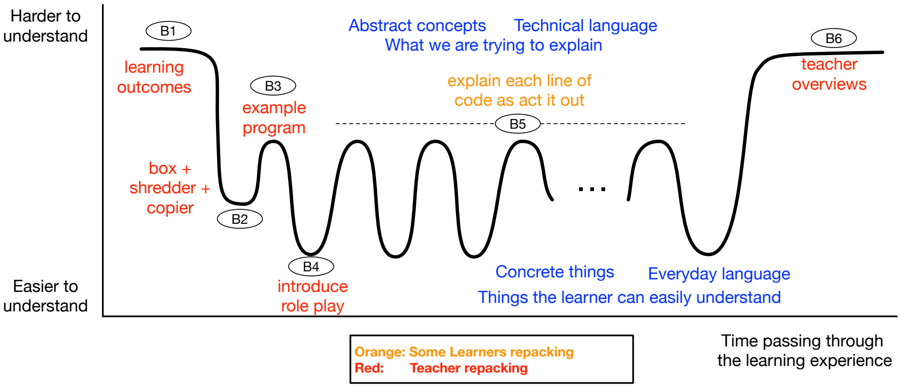

Figure 6: The semantic profile of the original Box Variable lesson plan. B5 represents all the waves between B4 and B6

- **(B3)** Introduce an example ('swap') program.
- **(B4)** Introduce the role play acting out the execution of the program.
- **(B5)** Explain each line of code in turn as it is acted out.
- **(B6a)** Learners individually summarise their understanding of variables and assignment in writing.
- **(B6)** The teacher summarises lessons about the meaning of variables and assignment and their execution.

6.4 Further Reflection

The analysis emphasises the need to give more repacking opportunities to the learners during the role play itself. Though this was not explicitly written into the lesson plan, the presenter did do this to some extent in the presentation by providing some opportunity for questions as each line of code was role-played and at the end while the volunteers were still there. An advantage of unplugged role play activities is that by making the abstract tangible, questions can be asked by pointing to the physical people acting things out. This helps learners who do not yet have mastery of the language to ask questions about what they do not understand. A physical role play means questions can be asked in simple language about physical, tangible experiences (low on the semantic profile). Answers can then support the creation of semantic waves by linking those questions to more complex, technical knowledge (shifting upwards). Making this step explicit in the lesson plan is therefore important, especially if the intention is that other teachers deliver the lesson based on the plan.

This leads to a new lesson plan for future delivery (Figure 8).

- **(B1)** Set out the learning outcome.
- **(B2)** Explain the variable as a metaphorical box with integrated shredder and copier.
- **(B3)** Introduce an example ('swap') program.
- **(B4)** Introduce the role play.

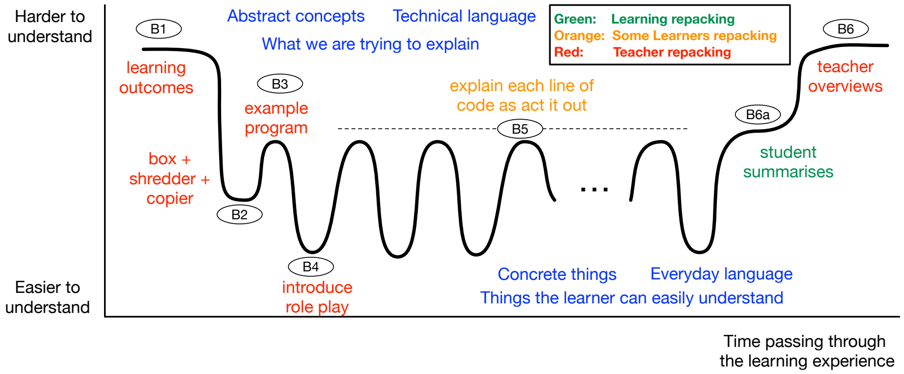

Figure 7: Semantic profile for the improved Box Variable lesson plan as delivered.

- **(B5a)** Explain each line of code in turn as it is acted out, giving opportunities to ask questions at each step.
- **(B5b)** Allow learners to ask questions about the whole activity/program.
- **(B6a)** Learners individually summarise their understanding of variables and assignment in writing.
- **(B6)** The teacher summarises lessons about the meaning of variables and assignment and their execution.

Further improvement could be added to make the repacking activity stronger. For example, learners could be asked to read the summary answer of a neighbour and look for differences to their own before discussing them. A further possibility would be to add a further Q&A session at the end, starting a further wave in explanations through packing and unpacking. Learners could also possibly copy the role play action as it happens moving paper between boxes.

One issue remains if this activity is considered alone. The "Box Variables" activity is working with a series of complex concepts (variables, values, assignment, lhs, rhs, sequences, ...) in a single activity. While there are a series of mini waves at the core of the activity, there is only minimal repacking by learners, even in the modified version. More repacking activity would be beneficial. However, there is limited opportunity within the activity to express a holistic overview. This analysis therefore suggests the need for onward activities that involve learners doing more direct and deeper unpacking and especially repacking. In short, the overall wave profile of the activity would benefit from being followed by subsequent waves in which learners are fully engaged in packing. Many such (more traditional) activities are possible. The remainder of the lesson, beyond the unplugged activity analysed here, did introduce further programming activities. Learners were next asked to do 'dry run' activities directly linked to the box metaphor, stepping through similar program fragments on paper (and in non-teacher CPD contexts, write programs). In the full multi-session version delivered to undergraduates, this was followed by a diagnostic test, and specific interventions for learners the test showed had the wrong mental model for the execution of the constructs. While it is beyond the scope of this paper to analyse these subsequent activities, the analysis suggests that unplugged demonstrations are more effective if followed by activities that generate waves in which learners are encouraged to actively connect what they are learning with the technical knowledge of computer programming.

6.5 Summary of the case study

The advantages of the final version over the original, in terms of offering a potentially more fruitful learning activity, are:

- The learners are encouraged to do repacking within the role play, helping them gain a deeper understanding of the sub-concepts in the context of the role play, before the final repacking.
- The additional final repacking step directly engages the learner in doing the repacking of concepts themselves, rather than passively listening to the teacher doing so.
- If extended with further waves based around learners drawing dry run tables repeating the role play on paper, then the learners will gain deeper technical mastery of the concepts than in the unplugged activity alone.

7 EXAMPLES OF UNPACKING AND REPACKING IN COMPUTING

We now summarise some strategies for unpacking and repacking in a computing context, extracted from these case studies. This is not intended to be exhaustive.

Unpacking examples (downwards moves) we have seen include:

- Making an explicit link between a learning outcome and an activity.
- Breaking down a main concept into a series of sub-concepts.
- Using metaphors (like a box to describe a variable).
- Linking a line of code to a role play activity.
- Linking a line of code to a physical object in an unplugged setting.

Repacking examples we have seen include:

- The teacher summarising main points.
- Asking learners to write a summary of main points.

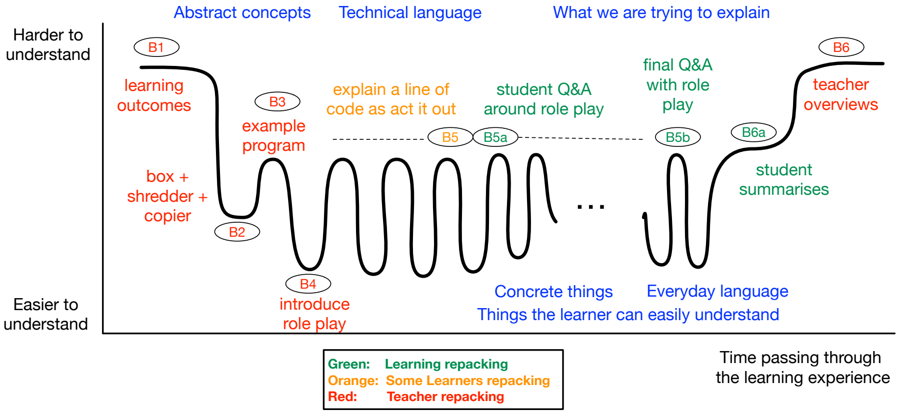

Figure 8: The semantic profile of the final plan of the Box Variable activity for future presentation.

- Asking learners to explain the abstract concept (including an example).
- Linking a metaphor back to the technical concepts.
- Encouraging questions about role play activity as a platform for shifting up into more technical ideas.
- Doing dry run exercises about a programming construct (or more generally doing follow on pencil and paper exercises directly linked to an unplugged activity).
- Linking action that has just happened in a role play activity to a line of code or concept.
- Linking a physical object used in an unplugged setting to a line of code or concept.

8 CONCLUSIONS AND FURTHER WORK

Semantic waves have been shown in other disciplines to provide a powerful way to think about the effectiveness of a learning activity as well as its delivery, and so to support the improvement of teaching. In previous work [34], we created a single semantic profile for an unplugged computer science lesson activity to evidence that this was possible. In doing this we reflected on our increased understanding of the mechanics of the lesson and suggested this understanding might help us improve lessons in general. In our work presented here on two CPD case studies, we have profiled further unplugged computer science activities and taken a step further. Using our three questions, we reflect on the heuristic profiles produced to identify opportunities for improvement revealed by the profile, and we have changed the activities including re-drawing the new profile.

Our contribution is the development and first time application of our three questions, which provide a framework for analysis of teaching activities, either CPD or student facing lessons. The analysis is quick and simple to do. It can be used in lesson planning to improve a plan before delivery, analyse the delivery itself, and to help educators reflect on improvements post-delivery. It can be applied in differing contexts: we applied it to both an activity delivered in a physical face-to-face workshop and an online delivery. We used a broad-brush version of semantic profiles. Finer granularity of analysis could be used to analyse critical parts of a lesson in more detail. An even broader version could look across multiple lessons. The approach helps explain how and why unplugged activities can be an effective form of teaching computing and how to improve them with follow-on activities.

We have delivered various workshops to in-service teachers explaining the theory and practice of semantic waves. We ask attendees to draw the semantic profiles for various activities we present, before discussing the profiles they have drawn. Attendees were able to draw semantic waves, though precise details of transitions differed. Doing so led to participant discussion about how activities could be improved. Participant feedback was very positive about the approach. We obtained feedback by a short questionnaire after the online CPD session containing the teleporting robot activity described. This session was, in fact, about semantic waves and involved participants attempting to draw the profile of the teleporting robot activity. One question asked "This lecture was…", with a choice of five options on a Likert scale: "very useful, useful, neutral, not very useful, not at all useful". All those providing feedback (n=17) chose "very useful" (n=12) or "useful" (n=5). Written comments also suggested several teachers explicitly intended to change their teaching as a result of the workshop and/or to use semantic profiles in their own lesson planning.

We subsequently delivered the session containing the Box Variables activity in an online context to teachers as part of a CPD session, including improvements as described. Feedback about the activity was very positive. All who provided feedback (n=19) chose "very useful" (n=17) or "useful" (n=2). Written feedback comments were also highly positive.

Our informal experiments as part of CPD workshops suggest that teachers can draw heuristic semantic profiles and that they can, and would like to, apply the approach to their own teaching. Of course, more formal studies around this are needed, but we believe the approach to be promising. We propose that the theory and analysis techniques of semantic profiling should become a standard part of teacher training for computing teachers both for initial teacher training and as part of continuing professional development.

Our case studies have involved a theoretical analysis of unplugged activities. Experimental studies are needed to compare the effects on learning of different kinds of waves in the context of computing. We are applying the approach to other kinds of learning experience, including written explanations, and other kinds of online and blended learning activities, as well as teaching across multiple sessions. One area we are looking at, for example, is the semantic profile of PRIMM [30].

A second line of enquiry is to evaluate the frequency with which teachers repack learners' knowledge, rather than the learners doing this for themselves. There are opportunities to link this to the development of a learner's useful and effective mental model of a computer and the role of constructivism in creating this [3, 4].

We have focused here on pure unplugged activities. To gain a deep understanding of programming concepts, learners need of course to also be experienced in technical programming contexts, developing technical programming skills in parallel with developing understanding. Doing so involves further waves of activity, for example doing tracing and programming activities. The activities presented are delivered as part of wider contexts. It was beyond the scope of this paper to analyse these wider contexts, but analysing the wave structure of such combinations is important further work. Having developed our methodology on familiar activities, we plan also, in next steps, to further validate the method on more general lesson plans and with inexperienced analysts.

Acknowledgements

This work was supported by the Institute of Coding, which is supported by the Office for Students (OfS).

REFERENCES

[1] Lawrence Barsalou, Paula Niedenthal, Aaron Barbey, and Jennifer Ruppert. 2003. Social embodiment. *Psychology of learning and motivation* 43 (2003), 43–92.

[2] Tim Bell, Jason Alexander, Isaac Freeman, and Mick Grimley. 2009. Computer Science unplugged: school students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology* 13, 1 (2009), 20–29.

[3] Mordechai Ben-Ari. 1998. Constructivism in Computer Science Education. *SIGCSE Bull.* 30, 1 (March 1998), 257–261. https://doi.org/10.1145/274790.274308

[4] Mordechai Ben-Ari. 2001. Constructivism in computer science education. In *Journal of Computers in Mathematics and Science Teaching*, Vol. 20. Assn for the Advancement of Computing in Education, 45–73.

[5] Margaret A.L. Blackie. 2014. Creating semantic waves: using Legitimation Code Theory as a tool to aid the teaching of chemistry. *Chemistry Education Research and Practice* 15, 462 (2014).

[6] Heather Bort and Dennis Brylow. 2013. CS4Impact: Measuring Computational Thinking Concepts Present in CS4HS Participant Lesson Plans. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (Denver, Colorado, USA) *(SIGCSE '13)*. Association for Computing Machinery, New York, NY, USA, 427–432. https://doi.org/10.1145/2445196.2445323

[7] Estella Williams Chizhik and Alexander Williams Chizhik. 2018. Using Activity Theory to Examine How Teachers' Lesson Plans Meet Students' Learning Needs. *The Teacher Educator* 53, 1 (2018), 67–85. https://doi.org/10.1080/08878730.2017.1296913

[8] Paul Curzon. 2014. The Box Variable Activity. Retrieved May 28, 2020 from https://teachinglondoncomputing.org/resources/inspiring-unplugged-classroom-activities/the-box-variable-activity/

[9] Paul Curzon. 2014. The Teleporting Robot (And Melting Snowman) Activity. Retrieved May 28, 2020 from https://teachinglondoncomputing.org/resources/inspiring-unplugged-classroom-activities/the-teleporting-robot-activity/

[10] Paul Curzon. 2019. Follow Semantic Waves, Tip 9 of Learning To Learn To Program. Retrieved May 21, 2020 from https://teachinglondoncomputing.org/learning-to-learn-to-program/ An informal blog on practical ideas about teaching programming.

[11] Paul Curzon and Shuchi Grover. 2020. Guided Exploration for Introducing Programming Concepts through Unplugged Activities. In *An A to Z handbook on teaching programming*, Shuchi Grover (Ed.). Chapter 7.

[12] Paul Curzon and Peter McOwan. 2008. Engaging with Computer Science through magic shows. *ACM SIGCSE Bulletin* 40, 3 (2008), 179–183. https://doi.org/10.1145/1597849.1384320 Also in Proceedings of ITiCSE 2008.

[13] Paul Curzon, Peter McOwan, Quintin Cutts, and Tim Bell. 2009. Enthusing and inspiring with reusable kinaesthetic activities. *ACM SIGCSE Bulletin* 41, 3 (2009), 94–98. https://doi.org/10.1145/1595496.1562911

[14] Paul Curzon, Peter McOwan, James Donohue, Seymour Wright, and William Marsh. 2018. Teaching Computer Science Concepts. In *Computer Science Education Perspectives on Teaching and Learning in School*, Sue Sentance, Erik Barendsen, and Carsten Schulte (Eds.). Bloomsbury Publishing, Chapter 8.

[15] Paul Curzon, Jane Waite, and Karl Maton. 2020. Semantic waves: analysing the effectiveness of computing activities. Retrieved May 28, 2020 from https://www.raspberrypi.org/app/uploads/2020/04/Cambridge-Computing-Education-Research-Symposium-Booklet.pdf Abstract presented at Cambridge Computing Education Research Symposium.

[16] Elena Lambrinos. 2020. *Building Ballet: developing dance and dancers in ballet.* Ph.D. Dissertation. School of Social and Political Sciences, University of Sydney, Sydney, Australia. Retrieved May 28, 2020 from https://ses.library.usyd.edu.au/handle/2123/22101

[17] Catherine Lewis. 2002. Lesson study: A handbook of teacher-led instructional change. *Research for Better Schools* (2002).

[18] Dorian Love. 2016. *Any tool works if you are using the language: The role of knowledge in ICT integration in a Johannesburg private school.* Master's thesis. School of Education, University of the Witwatersrand, Johannesburg, South Africa. Retrieved July 27, 2019 from http://wiredspace.wits.ac.za/handle/10539/22614

[19] Dorian Love. 2019. The grand tour: LCT informing pedagogy and instructional design. Retrieved May 31, 2020 from https://lct3.sched.com/event/QKZv/the-grand-tour-lct-informing-pedagogy-and-instructional-design Presentation presented at LCT3: Third International Legitimation Code Theory Conference.

[20] Dorothee Loziak, Peter McOwan, and Cristna Olivoko. 2016. The book of science mysteries. Version2.0. TEMI. Retrieved May 28, 2020 from https://cs4fndownloads.wordpress.com/temi-book-of-science-mysteries/

[21] Karl Maton. 2013. Making semantic waves: a key to cumulative knowledge-building. *Linguistics and Education* 24, 8-22 (2013).

[22] Karl Maton. 2014. *Knowledge and Knowers: Towards a realist sociology of education.* Routledge, Milton Park, Abingdon, Oxon.

[23] Karl Maton. 2020. Semantic waves: Context, complexity and academic discourse. In *Accessing Academic Discourse: Systemic functional linguistics and Legitimation Code Theory*, J. R. Martin, K. Maton, and Y. J Doran (Eds.). Number 59-85. Routledge, London, Chapter 3. https://doi.org/10.4324/9780429280726

[24] Karl Maton and Yaegan J. Doran. 2017. Semantic density: A translation device for revealing complexity of knowledge practices in discourse, part 1 - wording. *Onomázein* (March 2017), 46–76.

[25] Sharan B. Merriam. 2009. *Qualitative Research: A Guide to Design and Implementation.* John Wiley & Sons, San Francisco, CA.

[26] Marnel Mouton and Edward Archer. 2019. Legitimation code theory to facilitate transition from high school to first-year biology. *Journal of Biological Education* 53, 1 (2019), 2–20. https://doi.org/10.1080/00219266.2017.1420681

[27] Brandon Rodriguez, Stephen Kennicutt, Cyndi Rader, and Tracy Camp. 2017. Assessing computational thinking in CS Unplugged Activities. In *Proceedings 2017 ACM SIGCSE Technical Symposium on Computer Science Education.* ACM, New York, 501–506. https://doi.org/10.1145/3017680.3017779

[28] Richard M Ryan and Edward L Deci. 2000. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist* 55, 1 (2000), 68.

[29] Sue Sentance and Andrew Csizmadia. 2016. Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies* (2016), 1–27. https://doi.org/10.1007/s10639-016-9482-0

[30] Sue Sentance, Jane Waite, and Maria Kallia. 2019. Teaching computer programming with PRIMM: a sociocultural perspective. *Computer Science Education* (2019), 1–41. https://doi.org/10.1080/08993408.2019.1608781

[31] R.E. Stake. 1995. *The art of case study research.* Sage, Thousand Oaks, CA ; London.

[32] Renate Thies and Jan Vahrenhold. 2016. Back to school: Computer Science unplugged in the wild. In *Proceedings 2016 ACM Conference on Innovation and Technology in Computer Science Education.* ACM, New York, 118–123. https://doi.org/10.1145/2899415.2899442

[33] Jane Waite. 2017. Pedagogy in teaching Computer Science in schools: A Literature Review (After The Reboot: computing education in UK Schools). The Royal Society. https://royalsociety.org/~/media/policy/projects/computing-education/literature-review-pedagogy-in-teaching.pdf

[34] Jane Waite, Karl Maton, Paul Curzon, and Lucinda Tuttiett. 2019. Unplugged Computing and Semantic Waves: Analysing Crazy Characters. In *Proceedings of the 1st UK & Ireland Computing Education Research Conference* (Canterbury, United Kingdom) *(UKICER)*. Association for Computing Machinery, New York, NY, USA, Article 4, 7 pages. https://doi.org/10.1145/3351287.3351291