

# Extending the SOLO Model for Software-Based Projects

**Ilana Lavy**

Department of Management Information Systems, Yezreel Valley College (YVC) Israel  
Email: [ilanal@yvc.ac.il](mailto:ilanal@yvc.ac.il)

**Aharon Yadin**

Department of Management Information Systems, Yezreel Valley College (YVC) Israel  
Email: [aharony@yvc.ac.il](mailto:aharony@yvc.ac.il)

**Abstract**—In the process of assessing learning outcomes, educators use constructive tools for evaluating students' understanding and performance. In the present study MIS students were engaged in a full life cycle project as part of a Software Analysis and Design workshop. For evaluating their performance, we used the SOLO (Structure of the Observed Learning Outcomes) taxonomy. However during the various stages of the workshop we encountered some inherent limitations of the taxonomy that led us to the understanding that the SOLO taxonomy should be enhanced. This paper elaborates on these missing but required enhancements.

**Index Terms**—Higher education, peer assessment, the SOLO taxonomy, student perceptions, student performance.

## I. INTRODUCTION

Software systems are amongst the most complex systems developed by humans. Many tools and methodologies have been developed to cope better with this complexity. Agile software development methodologies, for example, stress rapid iterations of small and frequent releases with direct user involvement in the development process. The frequent releases reduce complexity while the user involvement ensures the development's outcomes. One of the widely used agile methodologies is XP (Extreme Programming), which concentrates on the development process rather than the managerial aspects of the process. Another agile methodology is Scrum, which is built to support changing development environments in which the requirements are not fully known at the start of the project, and even if known, these requirements could rapidly change during development.

Although agile methodologies seek constant realignment of the software development processes to the customer's requirements and needs, there are cases when the traditional methodologies are more successful, for example, when developing a package, when customer involvement cannot be secured or when traceability is required [1].

A widely known methodology for large projects' development is Structured Systems Analysis and Design, which provides a road map for the development activities required to ensure a successful project. Due to its importance, this methodology is a mandatory course as part of the Management Information Systems (MIS) curriculum. In our case, the course is taught as a workshop that outlines the hierarchical ordering of steps used to manage the development complexity. In addition, one of the aims of this workshop is to provide students with non-technical knowledge areas, such as critical thinking, interpersonal skills, team skills, and business understanding.

The workshop served as a framework within which students could demonstrate and augment their understanding of the ways technology usage can develop new organizational processes and achieve organizational goals. Taking account of the students' difficulties regarding these non-technical knowledge areas, the workshop structure employed many team-based activities and assignments. In addition to the ordinary technical assignments, such as planning, analyzing, and designing the project, the students were engaged in evaluating their fellow students' projects, which is a type of formative assessment.

As in the ordinary software development life cycle, the workshop assignments' complexity level rose incrementally, along with the cognitive skills needed to accomplish them successfully. To trace the students' progress we employed the SOLO (Structure of the Observed Learning Outcomes) taxonomy [2]. However, during the various phases of the project we became aware of some inherent limitations related to our course structure. Since the structure resembles a real life software development project' it led us to the understanding that the SOLO taxonomy should be extended so that it can be related to all of the project phases.

In this paper, we provide a theoretical background to the Bloom and SOLO taxonomies, describe the workshop structure, the limitations of the existing SOLO taxonomy, and include our suggestion to extend it. These newly suggested hierarchal levels augment the existing

taxonomy and provide a mechanism for assessing the learning outcomes associated with the elevated complexity of the whole development project.

## II. THEORETICAL BACKGROUND

In this section, we present a brief literature survey concerning the SOLO taxonomy – mapping levels of understanding.

### A. The SOLO taxonomy – mapping levels of understanding

In the research literature, there are several taxonomies by which learning processes and levels of understanding are classified [3, 4].

The Bloom taxonomy that was originally published in 1956 defines a six-level hierarchy for cognitive understanding: Knowledge Comprehension Application Analysis Synthesis Evaluation [4]. This understanding represents the process of learning, from simple remembering information to defining and creating something new and being able to evaluate and assess new knowledge. Although Bloom's taxonomy has been used extensively; it has experienced some criticism that it oversimplifies the nature of thought by assuming that cognitive processes are represented on a single, non-overlapping dimension [5]. For that reason, during the 1990's a team of researchers led by Lorin Anderson (a former Bloom's student) has suggested improving the taxonomy. The new proposed taxonomy still defined a six-level hierarchy; however, the terminology has changed. Instead of nouns, each level was referred to by the verb: Remember, Understand, Apply, Analyze, Evaluate, and Create. In addition, the newer taxonomy interchanged the order of synthesis (create) and evaluation (Evaluate). The reason was based on the understanding that creativity represents a higher level of complexity compared to critical thinking. One can criticize an idea without being creative, while creativity usually requires a degree of critical thinking. For addressing the one-dimensional structure of the original taxonomy, the revised taxonomy used a two-dimension matrix. One dimension for the six-level hierarchy, and the second for the results of the thinking, or forms of knowledge (Factual, Conceptual, Procedural and Metacognitive).

Another approach was suggested by Biggs and Collis, who developed a system for classifying the quality of students' work, known as the SOLO taxonomy [2]. The main advantage of the SOLO taxonomy, in relation to other educational hierarchies, and especially to original Bloom taxonomy, is its generality: it is not content-dependent, making it useable across a number of subject areas among different levels of students and on different types of assignments [3, 6, 7]. Several researchers have credited the SOLO taxonomy for its comprehensiveness and its ability to measure students' cognitive achievements [8, 9, 10]. Furthermore, some researchers found that SOLO is a useful framework for testing programming and Information Systems students skills [8,

11], and for that reason it was chosen for this study. The SOLO taxonomy has five levels of understanding that can be encountered in learners' responses to academic tasks [2]:

- Pre-structural – the task is not accessed appropriately and/or the student has not understood the task;
- Uni-structural – one or several aspects of the task are picked up and used (level of understanding is nominal);
- Multi-structural – several aspects of the task are learned but are treated separately. The student still lacks the “full picture” (understanding is equivalent to knowing about);
- Relational – the task's components are integrated into a coherent whole, with each part contributing to the overall meaning (understanding in the form of appreciating relationships);
- Extended abstract – the integrated whole at the relational level is re-conceptualized at a higher level of abstraction, which enables it to be generalized to a new topic or area. The integrated whole derived at the previous level is conceptualized at a more abstract level so that it can be used in different settings (understanding in the form of transferring concepts and involving metacognition).

The SOLO taxonomy has been used fruitfully both to classify students' work and to identify approaches used in the area of learning course material in post-secondary school settings. For these reasons, this research utilized the SOLO taxonomy to assess students' levels of learning. We used the SOLO taxonomy due to the objective criteria that it provided for measuring students' cognitive abilities [8].

## III. THE STUDY INFRASTRUCTURE

In this section, we present data regarding the study participants and the workshop which served as the infrastructure for our study. We also present information regarding the assignments given during the workshop and their timetable. Finally, we present information regarding the learning process evaluation methodology and its limitations, which were observed as part of the higher level of the workshop.

### A. About the study participants

Our research took place in the MIS department of a small regional college as part of the systems analysis and design (SAD) workshop, whose general objectives are to prepare the students for their final project and the real-world challenges they will face. The workshop is a mandatory course taken during the third (and last) year of their studies. At this stage, the students have acquired a reasonable understanding of the technical knowledge areas required for the workshop (software engineering, software modeling, UML (Unified Modelling Language)

usage, the Java programming language, Management Information Systems concepts, database design principles, etc.); however, even at that stage of their studies, most students still lack the non-technical knowledge areas (such as critical thinking and abilities to provide meaningful, helpful, and constructive feedback). For enhancing the students' soft skills, as defined by the IS 2010 curriculum guidelines for Undergraduate Degree Programs in Information Systems [12], the workshop augments knowledge and understanding gained in current and previous courses with practical, 'hands-on', and team-based study approach. Each team consists of four students that have to work collaboratively on their assignments. During the first lecture, the students were required to form the teams for the duration of the class.

### B. The course

Although the students were acquainted with all of the required technical knowledge, they were still not engaged in a whole process. The SAD workshop provides a simulation for all the development stages from the project's initiation to developing a prototype of the proposed system. This prototype is based on several incremental assignments and represents the main goal of the workshop. As part of the team-based assignments, each team received and worked on its own project. Each project included a general description of a virtual customer and business case. The students were asked to study their project description, address the problems presented in the business case, and suggest ways (and a software-based system) to solve these problems and achieve the customer's goals (which in many cases were only vaguely defined). To simulate real-world situations, the instructor assumed the role of the customer and answered all questions raised by the students. The workshop structure was based on incremental assignments that followed the software development life cycle. Each assignment was based on the previous one, and when combined together these assignments formed the whole project.

### C. Team assignments

Three types of team assignment were included in the workshop: (1) compiling four documents; (2) reviewing four documents (which were prepared by other teams); (3) preparing and delivering a class presentation.

- Compiling the documents – During the workshop the students were asked to submit four documents on (1) project initiation and planning; (2) system analysis; (3) system design; (4) system implementation. These documents describe the various stages of the project and represent knowledge, required for the project's advancement. Each of these documents had to follow, a template which was provided in advance and posted on the workshop website. In addition, for each template, consistent grading guideline was provided. These guidelines outlined the relative grade assigned to each paragraph in the document. The students were

asked to consider the various issues related to their project and to debate among themselves during the preparation of each document and then to present the solution agreed upon by their team.

- Reviewing documents - After a document was handed in, it was reviewed and assessed by another student team based on the document template and grading guidelines that were provided. The team-based peer review requires good communication skills, including the ability to give and receive constructive criticism, and above all, technical understanding regarding business issues, information system benefits, and development methodologies. The review process started with individual reviews followed by a collaborative team meeting in which they were asked to reach an agreed assessment. In the process of reviewing documents prepared by different teams, the students were exposed to other and new possible solutions. The assumption was that the students' groups were competent enough to assess their peers' documents, since the documents followed a predefined template, which the students used for preparing their own document. A successful assessment represents the ability to provide helpful and constructive feedback, and it correlates to the extended abstract level in the SOLO taxonomy.
- Presentation - The presentation was a summary of all the teamwork performed and was based on the integration of all the documents compiled. The presentation started with a brief description of the virtual customer, the business case, and associated problems. The main part of the presentation was a description of the information system proposed as a solution and how it would address the problems identified. In addition, the presentation had to relate to the risks associated with the project, the expected benefits, the time frame, and preliminary cost estimates.

### D. The workshop grading scheme

Each submitted document was reviewed and graded twice: once by the instructor and once by another team. It should be noted that the reviews were carried out in parallel and were not influenced by each other. Both assessments and grading were performed based on the common grading guidelines available on the workshop website. Use of the grading template served to enforce habits of precise and thorough analysis of documents as well as to enhance the students' understanding regarding additional possible solutions to similar problems. Being a third-year course, it is assumed that all students have some understanding about the SAD process, so the grading is based on the content provided and not the issues to be addressed and were outlined in the template. In compiling, a document, the uni-structural, multi-structural and abstract levels of the SOLO taxonomy are involved. Uni-structural implies understanding only one or a few issues; multi-structural implies understanding

and addressing more issues, but not all of them and abstract implies addressing all issues. The review and assessment stage reflects the extended abstract level of the SOLO taxonomy since it provides the student with an understanding of different solution to the same problem. By reviewing their peers' solution, the team is exposed to a different integrated whole. This leads to a higher level of abstraction in which the previously accumulated knowledge and understanding can be used in different settings and create another, yet absolutely valid solution to a given problem.

#### IV. THE STUDY

The study was performed over five years (2008-2012). The number of students participating in the workshop varied between 40 and 50. Due to the workshop's structure and its diverse and complex assignments mechanism, a methodological process for assessing students' knowledge was required, and the SOLO taxonomy was chosen. Each team project (Fig. 1) can be described as a hierarchical structure that consists of four incremental stages (Project Initiation, System Analysis, System Design, and System Implementation). For a successful project that will address and solve the customer problems, the students had to address and complete all these four stages.

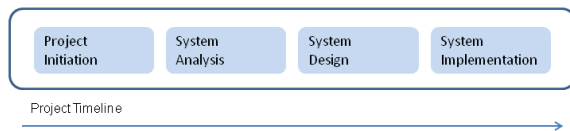


Fig. 1: Project stages

Each of these stages is represented by a relevant document that includes all the appropriate information required for completing the stage. However, due to the many issues involved in every stage, each document by itself can be described as a hierarchical structure that consists of many smaller tasks (Fig. 2).

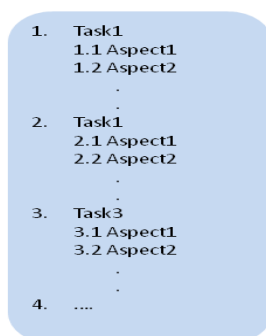


Fig 2: The document hierarchical structure

For example, to complete the Initiation document the students will have to address several tasks, each

represented as a paragraph of the document (Fig. 3). The Project Initiation stage, which sets the preliminary requirements for the project, has to be carefully thought through or else the analysis, design, and implementation stages will produce erroneous results.

1. Executive Summary
2. Customer Description
3. Current System Description
4. Current System's Problems
5. Preliminary Requirements
6. Feasibility Study
7. Preliminary project plan and staffing
8. Project Borders
9. Required Standards
10. Preliminary Risk Analysis
11. Recommendations

Fig. 3: The project initiation document

Other documents will address other issues applicable to the relevant stage of the project, which implies that the students will have to be involved with another set of relevant tasks. Each such task, for example, the customer description, is based on the students' understanding related to the aspects that are required and relevant for the task.

Based on the SAD assignments structure, the task to be performed is defined by one specific paragraph in a specific document. For example, the above-mentioned customer description paragraph represents one task to be performed. The document, on the other hand, will require the completion of all relevant tasks. For the Initiation stage, this means completing all 11 paragraphs of the Project Initiation document. Due to this structure, our interpretation of the five SOLO levels, when applied to the SAD workshop, is defined in Table 1.

This definition of stages and understanding levels as used in the workshop structure revealed some shortcomings related to the SOLO taxonomy. Like any other computing advanced course, the students in the workshop were required to address many hierarchical issues; however, the taxonomy provided a means of assessing the students' understanding only for the lower levels. Although the SOLO model expands the views of the problem in which first the students understand one issue, then several issues, then these issues are connected into a "whole" and at the last stage there are the meta-concepts that relate to this "whole". When addressing a large software project, as is the case with this workshop, the "whole" and the meta-concepts appear more than once. Each stage of the workshop is a "whole" with its meta-concepts, however, integrating several stages (documents) creates a larger "whole". These several levels of "wholes" and the appropriate meta-concepts are missing in the SOLO taxonomy.

Table 1: SOLO INTERPRETATION

Level	Explanation	Implication for the workshop
Pre-structural	The student lacks the ability to perform the task. There is insufficient understanding.	The student group does not have the proper understanding required for completing the task. Either the project is not clear or many of the underlying principles are still missing. For example, the students lack understanding of what is important and needed when describing the customer.
Uni-structural	One or several aspects of the task to be performed are taken into account. There is some understanding.	The student group understands some aspects related to a specific paragraph in the document but still lacks understanding of all of them. For example, a student group possesses an understanding of how to address some of the aspects required for describing the customer but not all of them.
Multi-structural	More aspects of the task are taken into account; however, the student still lacks the 'full picture'.	More of the aspects are clear, and the student group has started to address these aspects in compiling the answers. For some paragraphs, all required aspects are clear, while some other aspects are still not fully understood. This means that some, but not all of the document paragraphs have been completed.
Relational	All aspects are understood and integrated as a 'whole'. The student exhibits understanding of the parts as well as the relationships between them.	All aspects of all tasks to be performed are clear and are used for preparing the answers for all of the paragraphs in the document. At this stage, the knowledge level of student groups is sufficient for integrating all tasks into a coherent whole document.
Extended abstract	The 'whole' derived at the previous level is conceptualized at a higher level of abstraction so that it can now be used in different settings.	The student group has developed an abstract understanding of the steps and procedures required for compiling the document; the group understands other approaches that could be used to solve the same problem and is in a position to evaluate their strengths and weaknesses. This is done by assessing and evaluating other teams' solutions, which are represented by other teams' documents.

This definition of stages and understanding levels as used in the workshop structure revealed some shortcomings related to the SOLO taxonomy. Like any other computing advanced course, the students in the workshop were required to address many hierarchical issues; however, the taxonomy provided a means of assessing the students' understanding only for the lower levels. Although the SOLO model expands the views of the problem in which first the students understand one issue, then several issues, then these issues are connected into a "whole" and at the last stage there are the meta-concepts that relate to this "whole".

When addressing a large software project, as is the case with this workshop, the "whole" and the meta-concepts appear more than once. Each stage of the workshop is a "whole" with its meta-concepts, however, integrating several stages (documents) creates a larger "whole". These several levels of "wholes" and the appropriate meta-concepts are missing in the SOLO taxonomy.

To better visualize these shortcomings, we use the following figs (4-11) to depict the various levels of understanding as related to the workshop when utilizing the SOLO taxonomy.

#### A. The pre-structural level

The student demonstrates no understanding concerning the document to be compiled. He or she either does not understand the project or lacks the basic understanding of how information systems can solve the business problems described. Some typical questions related to the task to be performed (one specific paragraph in the template) that are required for assessing this level may include:

- What is the first stage in the project?
- What are the next stages?
- Who is the customer?
- What are his or her business problems?
- What can be done to better understand the business problems?
- How can we assess the system's contribution?

The student is at the pre-structural level if he or she cannot answer even one of these (and other) questions. This means that none of the aspects required for the various tasks are clearly understood. To better understand and visualize the various level of understanding, we will use a construction example in which these numerous

levels of understanding are mapped into the ability to build a house and a neighbourhood. Fig. 4 depicts this pre-structural level, of non-understanding by a blank rectangle (Tabula rasa). The rectangle represents the student's lack of understanding regarding the document to be compiled, and in the visualized example the student even cannot define the building blocks needed for the house. It should be noted that the shaded box is just the figure background and does not represent knowledge.



Fig 4: The pre-structural level

### B. The uni-structural level

The student is able to provide proper answers only to parts of the above-mentioned questions. These answers demonstrate only the beginning of the understanding needed to complete each of the tasks required in compiling the document. It is possible that for part of the above questions to which the student provided a proper answer, more profound questions are needed to be sure that the student knows both the underlying meaning and what has to be done in order to achieve the goal. Namely, after understanding the project to be developed, the student is able to address only some of the required activities or aspects of the problem. Usually these activities require additional questions that have to be explored and answered before the task can be successfully completed.

Furthermore, integrating all these understandings will still not provide the required solution. For example, for the Initiation document, after the student has understood the contribution of the system and how it can be measured, he or she will have to relate these factors to the feasibility study (how to perform it, what is needed, where the information will be obtained, etc.). Fig. 5 depicts this situation by referring to the construction example. The student understands that building a house requires several building blocks, but only some of these blocks appear in the chart (the pieces required for the roof and the building blocks needed for a window). Each such building block represents some aspects that have to be taken into account when compiling the document.



Fig 5: The uni-structural level

### C. The multi-structural level

The student is capable of completing at least one of the tasks required for the document. For example, relating to the Initiation document, the student will be able to assess the contribution of the information system related to:

- A comparison with the current situation,
- The new business potential,
- Benefits for the employees,
- The organizational knowledge and define measurable attributes for the contribution.

Fig. 6 depicts the multi-structural level of understanding. Relating to the construction example, at this level the student understands more aspects required for the task to be completed (constructing the house); nevertheless, he or she does not manage to complete it. The multi-structural level represents the understanding of more aspects (windows, door, roof, and walls), but without the ability to complete the task by building a house. The main difference between uni-structural (Fig. 5) and multi-structural (Fig. 6) is that uni-structural is related to unconnected pieces of knowledge. Multi-structural, on the other hand, represents larger pieces, each one made of smaller interconnected pieces, however, the multi-structural knowledge is not sufficient for the full document (the "whole"), or in the construction example the students understand how to build a wall or a roof but not the whole house. It should be noted that for completing each stage (or document in the workshop) the students have to fully understand all issues related to the document, as well as all the relationships among them. Fig. 6 depicts some higher level pieces required to build a house, however, the connections (relationships) are still missing.

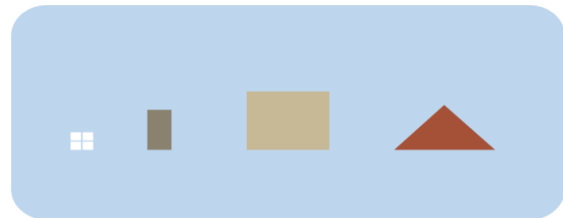


Fig 6: The multi-structural level

### D. The relational level

The student understands all the aspects required to complete all tasks and is able to integrate them into a coherent document (the "whole"). All these aspects lead to an orderly list of activities to be performed in order to compile the document. Namely, the student understands and is able to answer all previous questions and in addition other, more detailed ones. For example, if as part of the Initiation document the student is required to provide a preliminary work plan and schedule, this means he or she will be required to:

- Understand how to build such a plan and schedule,
- Define all tasks to be performed,



- Assess the preliminary resources needed for the project,
- Define the required capabilities,
- Set the order of events, and
- Draw a Gantt chart.

Fig. 7 depicts this relational level. All construction components have been integrated into the house.



Fig 7: The relational level

*E. The extended abstract level*

This is the highest level of understanding in the SOLO taxonomy and is represented in the workshop by two activities. The first is a document template, which includes a list of topics to be addressed. The student has to evaluate each one, assess whether it is relevant to the project at hand, and, if it is, define the missing but required information, and the tactics required to obtain it. The project described as part of the workshop is very generic, which means that the students have to apply the principles learned to find the missing pieces. The second activity that provides insight into the students' extended abstract level is the evaluation they perform. Only at this level are the students able to appreciate other valid approaches to similar problems, namely by assessing other documents that suggest different ways of solving the problem. Fig. 8 depicts this situation. In addition to the house constructed by the student, he, or she is able to appreciate other houses (solutions) proposed by other teams. All solutions are valid, and each one has its strengths and weaknesses. At this level of understanding, the students are knowledgeable and capable of assessing and criticizing each such solution. This level of understanding provides self-criticism, as was demonstrated by several teams that asked to change their solutions based on the knowledge and understanding they gained from assessing their peers' solutions.



Fig 8: The extended abstract level

This extended abstract level of understanding, which represents the highest level of the SOLO taxonomy, relates to the 'whole' (one document or when relating to

the visualization example, one house). However, when considering the project, it consists of four documents; when using the construction example it consists of a house, which represents the highest level, but is just one component of the neighbourhood. So when proceeding to the whole project (or the neighbourhood), an additional level of understanding is required but is, unfortunately, missing from SOLO. The current extended abstract level represents the multi-relational level in the project (or the neighbourhood).

*F. The multi relational level*

This is a proposed new level that in the workshop is represented by understanding one or more documents (project stages) but not all of them. For example, a student may understand all the tasks required for both the Initiation and Analysis documents but may be incapable of relating this knowledge to the Design document. Specifically, the student lacks understanding on how to define the proper database schema that is needed for the project.

Fig. 9 depicts this situation. In the construction example, each document is represented by a house. The student is capable of combining several houses to construct a partial neighbourhood, but it is not fully populated.

The level of understanding represented by this figure is of a student who has managed to complete one or more documents but is not capable of moving to the next level of completing several interrelated documents.



Fig 9: The multi relational level

*G. The extended relational level*

This is an additional proposed new level that in the workshop is represented by understanding the project as a whole. Students at this level have the required understanding to compile all documents and reach the higher level of integrating the documents into a coherent project.

Relating to the construction example, this level represents the understanding required to build the whole neighbourhood as depicted in Fig. 10.



Fig 10: The extended relational level

#### H. The extended abstracted abstract level

The ability to understand that a project can be implemented in various ways necessitates additional levels of understanding. We suggest naming it the 'extended abstracted abstract level'.

Fig. 11 depicts this understanding in relation to the visualized example. The figure contains three different neighbourhoods. This stage is represented in the workshop by the students' presentations and the evaluation performed by each student independently. This is the last stage in the workshop, in which each team has to present its solution. All other students have to assess the solution based on their understanding of the customer's problems and possible ways to address and solve these problems. Furthermore, the students have to evaluate the solution as proposed and its strengths and weaknesses.



Fig 11: The extended abstracted abstract level

Only after extending the taxonomy and adding the three levels was it possible to fully assess the teams' level of understanding. The original taxonomy stops at the extended abstract level. This relates to the team's understanding that basic concepts can be used in other settings, for example by suggesting different ways of implementing the first stage. However, a higher level of understanding is required for integrating the two stages of the project and realizing that there are several ways of performing these integrations. Thus the newly suggested multi-relational level defines a new larger and coherent whole that was integrated from previous stages; each one represents a smaller whole.

The next extended relational level is required to represent the understanding associated with an even larger whole – the integration of the three stages that define the whole project. The need for that level arose since some students may understand and correctly implement the integration of two stages, but fail to understand how the whole project has to be integrated. The last suggested level (extended abstracted abstract) is derived from the previous level and provides the understanding that the concepts that define the project can be used in different settings in defining new solutions to the same problem.

This understanding was demonstrated by reviewing and assessing the project submitted by another team. All these projects comprise three stages, and each stage represents the extended abstract in the SOLO taxonomy. However, for assessing the whole project the extended abstracted abstract level is required.

#### V. DISCUSSION AND CONCLUDING REMARKS

The SOLO taxonomy is intended for assessing learning outcomes as demonstrated in our case by a single assignment. When applying the SOLO taxonomy to a course like systems analysis and design, some limitations emerge. As a third-year course, the workshop integrates knowledge that students have accumulated during their previous studies. This means that the pre-structural level does not exist and all students demonstrate a higher level of understanding; furthermore, the next two levels are seldom present since the students have assimilated the basic understanding, as demonstrated by the fact that they successfully completed the previous courses. This means that for courses such as this workshop only the upper part of the taxonomy is being used. On the other hand, the workshop's structure revealed a shortcoming of the taxonomy.

In his *Soft Systems Methodology*, Checkland describes many systems that surround us [6], such as engineering systems, natural systems, and even software development systems. Each such system has a goal, is part of the environment, receives input and produces output, and is made up of integrated and connected components. Sometimes the components themselves are systems of a smaller scale.

Applying the SOLO taxonomy to each component provides insight into the learning outcomes of that component; however, it cannot be applied to the whole system. When applying the SOLO taxonomy to the workshop, the extended abstract level demonstrates complete understanding of one single assignment. For successfully compiling a document, that represent one stage of the workshop, the students have to be at the Relational level, in which all the ideas and knowledge are combined together to produce the "whole" (the document). Furthermore, due to the workshop's structure, in which the students have to assess their peers' documents, they have to be in the highest level of the taxonomy – the Extended abstract. At this level the students have to be able to understand other approaches to solve the problem they solved in their document.

However, although being at the highest level of understanding of the taxonomy, the workshop includes four assignments, where each depends on and enhances the previous one. Furthermore, to fully understand the workshop and appreciate its outcomes the students have to integrate all four documents into a coherent working prototype. Since the SOLO taxonomy does not define a higher level of understanding, this type of learning outcome cannot be evaluated by the existing SOLO taxonomy. A student may fully understand one document and even be able to evaluate other documents that define different approaches, but still fail to integrate several stages of the same project. Furthermore, since all engineering products can be defined as systems of systems [6], the SOLO taxonomy is limited in assessing the level of understanding of these higher systems.

Relating to the visualized example, a student may fully



understand how to build a house, or may be at the highest level of understanding according to the SOLO taxonomy, but this does not mean that he or she will be able to build a neighbourhood. Fig. 10 depicts a large system (a neighbourhood) that comprises the integration of several components (houses). Each component is a system by itself. The extended abstract level represents the understanding of each system (the house); however, the taxonomy lacks an additional level required to address this situation, i.e. to understand the integration of these components into a larger-scale system. Furthermore, as is the case with larger software development projects, this neighbourhood can be used as a component in an even larger system (a city). Our proposed taxonomy includes the new Multi-relational additional level that represents understanding of two stages (initiation and analysis). This new level is requested because the analysis is performed after the initiation as is dependent of the initiation outcomes and augments them.

The next proposed level to the taxonomy, the Extended relational is required to assess students' knowledge regarding the whole workshop. This knowledge is the result of integration all four documents (stages) into a working system. The difference between the Multi-relational and the Abstract relational is related to the magnitude of the integration. While the Multi-relational represent understanding of two documents integration, The Abstract-relational represents the understanding of all the projects' stages (documents).

The last level proposed, the Extended abstracted-abstract represents the students understanding that the whole project, although is integrated from four stages, can be developed in various ways. The students at this level understand and appreciate a different project while being able to assess its strengths and weaknesses.

The process may continue for several additional levels in which each newly formed system is just a component of the larger system. In our case the SOLO limitations were observed as part of the systems analysis and design workshop, since this course addresses the issue of software development complexity and the standard method of coping with this complexity is to analyse the system or dismantle it into its components. If the components are still too complex, each component will be dismantled further. This process of decomposition is repeated until the components are small, simple, understandable, and manageable. On the other hand, building the system from its components requires several levels of hierarchical understanding that extend the levels defined by the SOLO taxonomy.

One may suggest using the SOLO taxonomy as a sliding window in which similar levels of understanding are being used over the duration of the project. Specifically, as related to the systems analysis and design workshop, this means that when dealing with the first assignments the relational level will be applied to the document. When dealing with the whole project, however, the relational level will be applied not just to the first document, but to the project. Using this sliding window mechanism, which provides different meanings for the

same level of understanding, may overcome the limitation; however, since the knowledge required for the various levels is significantly different, this is a wrong direction.

With a single assignment, the current taxonomy is sufficient; however, with larger projects at least two additional levels are required. The extended abstract level of the first assignment is actually the multi-structural level, or may sometimes be just the uni-structural level, of the project or the higher-level system. On the other hand, very large project may require further level to be added to the taxonomy, if assessing understanding of the whole system is required (or possible).

Furthermore, one may criticize the paper for presenting a rather narrow view of the SOLO model and then the described limitations arise. In a broader view, the SOLO model defines (1) understanding of one or a few issues, then (2) understanding more issues, then (3) understanding the connections among all issues and at the last level (4) understanding the meta-concepts. However using this broader view represents another limitation since it means that in the workshop the relational level of understanding relates to the whole project and the multi-structural relates to just one stage. In this case, the SOLO model lacks the means to differentiate between the levels of understanding of students who successfully define one document and the ones that defined more than one document, since both are at the multi-structural level. In addition this approach assumes that the workshop stages are just steps of the project, which is only partially true. Besides being part of the whole project, each stage is a "whole" by itself for which all SOLO level of understanding should be applied.

#### REFERENCES

- [1] Griffiths, M. (2007). Agile suitability filters. Retrieved January 2013 from [http://leadinganswers.typepad.com/leading\\_answers/files/aggregate\\_suitability\\_filters.pdf](http://leadinganswers.typepad.com/leading_answers/files/aggregate_suitability_filters.pdf).
- [2] Biggs, J.B., & Collis, K.F. (1982). Evaluating the quality of learning: The SOLO taxonomy (structure of the observed learning outcome). New York: Academic Press.
- [3] Anderson, L.W., & Krathwohl, D.R. (Eds.) (2001). A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives. New York: Addison Wesley Longman.
- [4] Bloom, B.S. (1956). Taxonomy of educational objectives, the classification of educational goals – Handbook I: Cognitive domain. New York: McKay.
- [5] Furst, E. (1994). Bloom's taxonomy: Philosophical and educational issues. In L. Anderson & L. Sosniak, (Eds.), Bloom's taxonomy: A forty-year retrospective (pp. 28-40). Chicago: The National Society for the Study of Education.
- [6] Chan, C.C., Tsui, M. S., Chan, M. Y. C., & Hong, J. H. (2002). Applying the structure of the observed learning outcomes (SOLO) taxonomy on student's learning outcomes: An empirical study. *Assessment & Evaluation in Higher Education*, 27(6), 511-527.
- [7] Hattie, J., & Purdie, N. (1998). The SOLO model: Addressing fundamental measurement issues. In B. Dart & G. M. Boulton-Lewis (Eds.), *Teaching and learning in higher education*. Camberwell, VIC: Australian Council for Educational Research.

- [8] Chick, H. (1998). Cognition in the formal modes: Research mathematics and the SOLO taxonomy. *Mathematics Education Research Journal*, 10 (2), 4-26.
- [9] Lake, D. (1999) Helping students to go SOLO: Teaching critical numeracy in the biological sciences. *Journal of Biological Education*, 33(4), 191-198.
- [10] Van Rossum, E. J., & Schenk, S. M. (1984) The relationship between learning conception, study strategy and learning outcome. *British Journal of Educational Psychology*, 54, 73-83.
- [11] Lister, R., Simon, B., Thompson, E., Whalley, J. L., & Prasad, C. (2006). Not seeing the forest for the trees: Novice programmers and the SOLO taxonomy. In *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, Bologna, Italy, June 26–28, 2006. ITICSE '06. ACM Press, New York, NY, 118-122.
- [12] Topi, H., Valacich, J. S., Wright, R. T., Kaiser, K., Nunamaker, J. F., Sipior, J. C., & de Vreede, G. (2010). IS 2010: Curriculum guidelines for undergraduate degree programs in information systems. *Communications of the AIS*, 26(18).

#### AUTHORS' PROFILES



Ilana Lavy is an associate professor and the department head of Management Information Systems at the Max Stern Yezreel Valley College. Ilana's primary teaching areas are introduction to databases, data structures, introduction to management information systems, statistics and probability. Her PhD dissertation (in the Technion) focused on the understanding of basic concepts in elementary number theory. After finishing doctorate, she was a post-Doctoral research fellow at the Education faculty of Haifa University under the supervision of Prof. Anna Sfard. Her research interests are in the field of pre service and mathematics teachers' professional development as well as the acquisition and understanding of mathematical and computer science concepts. Prof. Lavy published over hundred papers and research reports.



**Aharon Yadin** is a Senior Lecturer at the Max Stern Yezreel Valley College (YVC), Management Information Systems Department. Aharon's primary teaching areas are computer architectures and Introduction to Computer Science, Software Analysis and Design and Projects Management. Prior to entering the academic world, Aharon worked in the High Tech industry. He has over 40 years of IT experience including: management, system performance analysis and enhancement, computer center and IS management, wireless networks and communication technologies and document management.

Aharon has published over one hundred papers, essays and scientific and technological reports (many in Hebrew), he is the author of 9 instructional books and consults the European Commission on software related projects and technologies.