

# Combining Symbolic and Sub-symbolic AI in the Context of Education and Learning

**Rainer Telesko, Stephan Jüngling, Phillip Gachnang**

FHNW University of Applied Sciences and Arts Northwestern Switzerland, School of Business  
School of Business, Riggensbachstrasse 16, CH-4600 Olten, Switzerland  
{rainer.telesko|stephan.juengling|phillip.gachnang}@fhnw.ch

## Abstract

Abstraction abilities are key to successfully mastering the Business Information Technology Programme (BIT) at the FHNW (Fachhochschule Nordwestschweiz). Object-Oriented (OO) is one example - which extensively requires analytical capabilities. For testing the OO-related capabilities a questionnaire (OO SET) for prospective and 1<sup>st</sup> year students was developed based on the Blackjack scenario. Our main target of the OO SET is to identify clusters of students which are likely to fail in the OO-related modules without a substantial amount of training. For the interpretation of the data the Kohonen Feature Map (KFM) is used which is nowadays very popular for data mining and exploratory data analysis. However, like all sub-symbolic approaches the KFM lacks to interpret and explain its results. Therefore, we plan to add - based on existing algorithms - a "postprocessing" component which generates propositional rules for the clusters and helps to improve quality management in the admission and teaching process. With such an approach we synergistically integrate symbolic and sub-symbolic artificial intelligence by building a bridge between machine learning and knowledge engineering.

## Introduction

OO-related content exists in a considerable number of BIT modules, mainly related to Business Analysis, Software Engineering and IT Architecture. Researchers generally agree that abstraction ability is a necessary skill for OO design and OO programming (Alphonse & Ventura, 2002, Bennedsen & Caspersen, 2008; Nguyen & Wong, 2001; Or-Bach & Lavy, 2004); however, a reliable instrument to test a person's level of abstraction ability in the context of OO has not yet been developed. We focus in our research on the abstraction ability, which is needed to build OO-related abstractions based on the understanding of a predefined domain.

## Motivation and Background

The FHNW is striving towards receiving an accreditation according to AACSB (Association to Advance Collegiate Schools of Business's) (AACSB, 2019). Earning an AACSB accreditation implies that a concrete framework to continually measure the quality of the school's programs has to be in place. Two core processes which ensure the quality of the student's target skills are the admission and teaching processes. The admission process verifies that students own the necessary entry requirements and skills for mastering the programme, and the teaching process should contain a monitoring component to measure the student's performance. The AI-based data analysis based on the OO SET is a valuable contribution to ensure the quality of these processes. The evaluation aims to identify clusters of students and to predict their potential performance in terms of passing the assessment stage.

AACSB requires comprehensible adjustments (system or content improvements) if the student's performance does not meet the requirements (Assurance of Learning process - AoL). For this reason, we opted for a 2-step data analysis method that adds an explanatory component to the sub-symbolic AI. All in all, the OO SET can be integrated into AoL to handle OO topics across modules. As the composition of the students - and thus the existing entry skills - change over time, an on-line monitoring together with a moving time window is necessary.

## Current State of Work

For setting up the OO SET we focus on the abstraction ability, which is relevant not only for the beginning of programming in the small (classes, attributes, relationships, hierarchies) but also for programming in the large (libraries, frameworks, design patterns, software architectures). OOA and OOD are still predominant in software engineering and

working with models as abstractions from code (e.g. UML class diagrams) is vital not only for software- but also for database-engineering (ERD).

The OO SET was implemented with Google Forms (<https://forms.gle/rj5NSqmgTth1dm2f7>). As a “test” domain, a scenario related to the card game Blackjack was selected, as this game is widely known and can be relatively easy explained for a short assignment like the questionnaire. Furthermore, the different cards and conditions (e.g. of a particular casino) offer various possibilities for tasks in connection with OO concepts. The first prototype contains 30 multiple choice questions, and in the first round of testing we had 27 participants. In order to get a clearer picture of the students’ aptitude and to support a more sophisticated evaluation, every question is assigned both to a OO concept category and a level according to Bloom (1956). In the BIT program both students with and without pre-knowledge in programming are enrolled. In order to “simulate” such a situation for the OO SET, two test groups (i.e. total beginners) and BIT 1st term students with some knowledge based on the running Programming module were considered.

The first part of the questionnaire asks for information about participants, such as age, prior OO knowledge, gender, etc. and covers a basic overview of relevant OO principles by using text, graphics and videos (see fig.1)

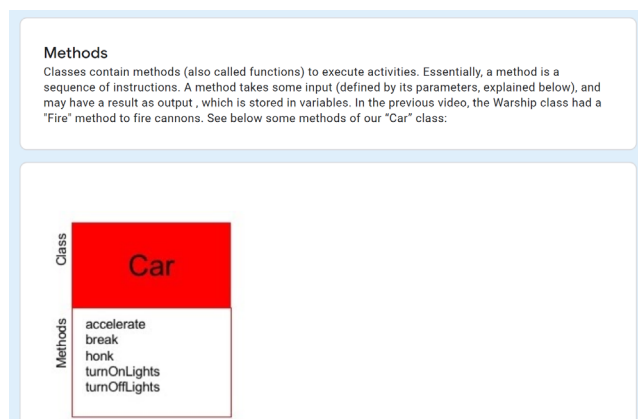


Figure 1– OO SET tutorial for methods

The second part of the questionnaire (see fig. 2) deals with questions related to core OO concepts. The selection of the OO concepts discussed and tested in the questionnaire was made in consultation with BIT lecturers and based on similar field research (Bennedsen & Schulte, 2006; Okur, 2007). The list of the OO concepts used and tested in the questionnaire includes: classes, objects, classes vs. objects, attributes, classes vs. attributes, methods in classes, parameters of methods, inheritance, multiplicity, encapsulation and relationships between classes (association, aggregation, composition).

While classes and objects are regarded as rather simple, encapsulation is seen as a more advanced concept; thus, the

concepts vary in complexity. However, elements such as polymorphism, abstract classes, interfaces, and Design Patterns that are classified as more complex (Bennedsen & Schulte, 2006; Okur, 2007) were not part of the questionnaire.

Figure 2– OO SET example questions

One key criterion for selecting the elements above was to have a high degree of overlap with the introductory programming module in BIT which is the major challenge for students to master successfully the assessment stage. Currently this programming module follows an OO-first approach using Eclipse as Integrated Development Environment and JavaFX as framework for programming Graphical User Interfaces. After deducting points for incorrect answers to questions where multiple selections were possible, the average questionnaire score across all student groups (i.e. based on their self-indicated level of OO knowledge) was 55%. Intuitively, students that indicated they had prior OO knowledge (i.e. identifying as “intermediate” or “advanced”) performed better than those with little or no prior OO knowledge, as can be seen in the following figure 3.

Bloom Question Level	Average Score (All students)	No knowledge / Beginner average score	Intermediate / Advanced average score
Remember	67%	65%	70%
Understand	52%	48%	60%
Analyze	37%	33%	43%
All levels	55%	52%	61%

Figure 3– Results from Bloom

Based on the results of the overall score from all Bloom levels being above 50%, and given that the scores increase with the students’ level of OO knowledge, the questionnaire can be considered as successfully testing the abstraction ability.

Test validity was checked by comparing the results of OO SET with exam results from a module covering abstraction abilities, namely “Introduction into BIT”. This module also belongs to the assessment stage.

## Data Mining using the KFM

Data from the OO SET can be used to optimize the admission process and to identify clusters of students with similar performance. Especially interesting are the students who fail the aptitude test because they might share common characteristics.

For our first experiments, we used the Kohonen Feature Map (KFM) (Kohonen, 1998; Oja & Kaski, 1999). The KFM is especially interesting when the clusters are not known in advance, as it is the case in the data related to the OO SET. The KFM is a two-layer, fully connected, feedforward network where a multidimensional input vector is mapped to a grid of output neurons. The KFM enables a topological preservation of input vectors on the output layer after training, i.e. input vectors with a high degree of similarity in terms of Euclidian distance metrics are mapped to neighbor neurons on the output layer. In our case the student metadata (gathered from part 1 of the OO SET questionnaire, like age, origin, entry qualifications etc.) is coded in the input vector. The number of the neurons in the competitive output layer is chosen arbitrary and builds the grid for student neighborhoods sharing similar characteristics.

The following figure 4 shows the U-matrix (unified distance matrix) of a trained KFM of the OO SET test results. Dark neurons define the dissimilarity to the neighbor neuron and the line highlights the borders of similarities in all dimensions of the input vectors, thus similarity clusters of the trained neurons.

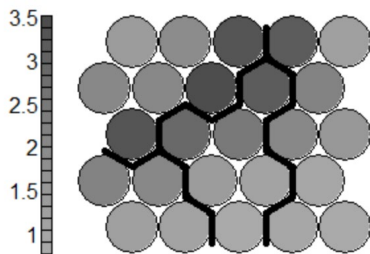


Figure 4 – U-matrix

A hierarchical clustering on the distances of the neurons to their neighbors has been used as shown in figure 5. The result are the three clusters colored in red, green and blue. These clusters are the result of the similarity in all dimensions of the neurons.

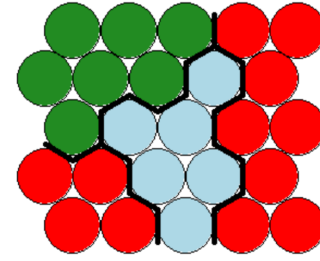


Figure 5 – Clusters

Certain dimensions are especially interesting, in order to derive conclusive rules for the OO SET. Fuzzy rules are especially interesting because they are understandable by humans and can easily be processed. One of the most interesting dimensions in the OO SET is whether the student passed the test or not. The heatmap in figure 6 shows the three clusters reduced on this dimension. The darker a neuron is, the more pronounced is its pass characteristic and the more contrast a neuron has to the neighbors the more dissimilar the neuron is on this characteristic.

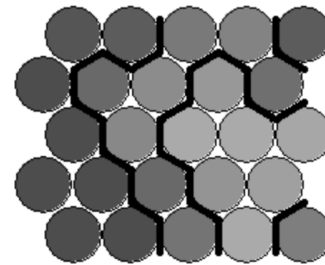


Figure 6 – Pass Clusters

The first cluster with the light colored neurons represents students, which clearly did not pass the test. It is surrounded by the second cluster, which groups students, which are close to the pass/fail borderline. The third cluster with the dark colored neurons groups the students who clearly passed the test. These clusters combined with the dimension weights, as shown in figure 7, allow to derive meaningful fuzzy rules based on the values in the different dimensions.

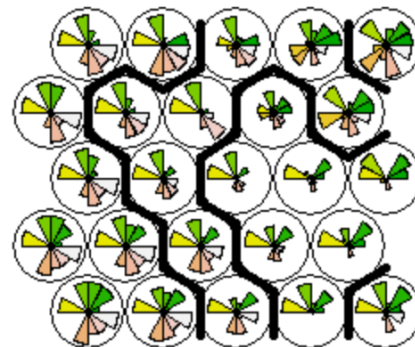


Figure 7 – Dimension Weights

The weights of each neuron are representative or similar to the student's characteristics mapped to that neuron. The correlation between age and education, the sections with the two darkest greens are significantly pronounced and correlate with the white colored pass dimension. This leads to a conclusive fuzzy rule, that students with a high education level and a high age will most likely pass the test.

The KFM uses a stochastic learning algorithm because input vectors are selected randomly in order to avoid a bias. This implies that with every experiment the feature map will look differently by preserving the major topological preservations. Another important issue is that the KFM does not provide clusters at the end of the learning algorithm. Every neuron on the output layer stands for a best-matching unit, in our case a student best matching his input vector. Neural networks only work well if sufficient amount of representative data is available which is not yet the case. Because we are currently spreading the OO SET among all beginning and prospective students this situation will clearly improve.

### KFM-based knowledge base

In summary, our approach is divided into four-steps. First, we set up a KFM based on the data gathered with the OO SET. The KFM itself does not provide any clusters. In order to get clusters, a post-processing is necessary in the second step. Our approach is based on "coloring" output neurons based on their distances on the map. Such a colored KFM (U-matrix) shows light neurons belonging to the same cluster with dark neurons at the cluster border (Ultsch & Korus, 1995). In the third step, these clusters can be additionally represented as fuzzy rules, which enables us to build up a KFM-based knowledge base (Ultsch & Korus, 1995; Malone, 2006). The administrative staff as well as the lecturers will use this knowledge base within the admission and teaching processes. However, these rules have to be continuously updated in the fourth and last step due to the probably changing student distribution and performance results in the OO SET over time.

With such an approach (starting with the KFM and ending up with rules) also the main drawback of sub-symbolic AI – which consists in not being able to explain its results – can be removed. The FHNW quality management gets a DSS (decision support system) which enables to pay attention to specific student groups during the enrollment and admission process.

### Conclusion

The main target of our research is to identify clusters of students which are likely to fail in the OO-related modules in

order to optimize the admission and teaching processes. Respective performance data concerning basic OO concepts is generated via a questionnaire (OO SET). For learning the "similarity" of students a neural network was used. As reported in the previous section, the use of KFM provides the possibility to cluster students with similar levels of understanding OO concepts. This allows deriving rules as well as subsequent learning units, which can be mapped to the particular needs of the different student clusters and be based on the taxonomy of Bloom. However, this process can be generalized to other disciplines throughout the course of studies. KFMs can be used to extract more general rules not only related to the abstraction capabilities and OO thinking. All our modules are described with learning goals, which are mapped to the different taxonomy levels of Bloom. Given the fact, that many lecturers already work with questionnaires, which are published in the learning management system Moodle, these questionnaires could easily be reused to analyzed with KFMs and provide general possibilities to analyze and individualize students the students learning paths. Educational guidance can be provided based on more detailed skill maps and support the overall process of AoL.

### References

- AACSB (2019, November 14). Association to Advance Collegiate Schools of Business. Retrieved from <https://www.aacsb.edu/>.
- Alphonse, C. & Ventura, P. (2002). Object Orientation in CS1-CS2 by Design. Proceedings of the 7th Annual Conference on innovation and Technology in Computer Science Education, Aarhus, Denmark, pp. 70-74.
- Bennedsen, J. & Schulte, C. (2006). A Competence Model for Object Interaction in Introductory Programming. In Proceedings of the 18th Workshop of the Psychology of Programming Interest Group, pp. 215-229.
- Bennedsen, J. & Caspersen, M. E. (2008). Abstraction Ability as an Indicator of Success for Learning Computer Science? Sydney, Australia: ICER'08.
- Bloom, B. S. (Ed.). (1956). Taxonomy of educational objectives: The classification of educational goals, by a committee of college and university examiners. New York: D. McKay.
- Kohonen, T. (1998). The self-organizing map. *Neurocomputing*, 21(1-3), pp.1-6.
- Malone, J., McGarry, K., Wermter, S. and Bowerman, C. (2006). Data mining using rule extraction from Kohonen self-organising maps. *Neural Computing & Applications*, 15 (1), pp. 9-17.
- Nguyen, D. & Wong, S. (2001). OOP in Introductory CS: Better Students Through Abstraction, Proceedings of the fifth Workshop on Pedagogies and Tools for Assimilating Object-Oriented Concepts, OOPSLA.
- Oja, E. and Kaski, S. (1999). *Kohonen Maps*. Elsevier, Amsterdam.
- Okur, M. (2007). Teaching Object Oriented Programming At The introductory Level. In *Journal of Yasar University*. 1 (2), pp. 149-157.

Or-Bach, R. & Lavy, I. (2004). Cognitive Activities of Abstraction in Object Orientation: An Empirical Study. *Inroads - the SIGSCE Bulletin*. 36 (2), pp. 82-86.

Ultsch, A. & Korus, D. (1995). Self-organizing Neural Networks for Acquisition of Fuzzy-Knowledge, *Proceedings of the 3rd GI-Workshop Fuzzy-Neuro-Systeme in Darmstadt*, pp. 326-332.