Classification of Shared Tasks Used in Teaching

Theresa Elstner
Leipzig University
Leipzig, Germany
theresa.elstner@uni-leipzig.de

Bärbel Hanle
Leipzig University
Leipzig, Germany
baerbel.hanle@informatik.uni-leipzig.de

Frank Loebe
Leipzig University, ScaDS.AI
Leipzig, Germany
frank.loebe@informatik.uni-leipzig.de

Maik Fröbe
Friedrich-Schiller-Universität Jena
Jena, Germany
maik.froebe@uni-jena.de

Nikolay Kolyada
Bauhaus-Universität Weimar
Weimar, Germany
nikolay.kolyada@uni-weimar.de

Janis Mohr
Bochum University of Applied Sciences
Heiligenhaus, Germany
janis.mohr@hs-bochum.de

Jörg Frochte
Bochum University of Applied Sciences
Heiligenhaus, Germany
joerg.frochte@hs-bochum.de

Sven Hofmann
Leipzig University
Leipzig, Germany
sven.hofmann@informatik.uni-leipzig.de

Benno Stein
Bauhaus-Universität Weimar
Weimar, Germany
benno.stein@uni-weimar.de

Martin Potthast
University of Kassel, hessian.AI, ScaDS.AI
Kassel, Germany
martin.potthast@uni-kassel.de

ABSTRACT

We report on our experience with shared-task-based teaching based on 12 undergraduate and master's courses. From the lessons learned in these courses, we derive a novel classification of shared tasks into four classes based on properties of their solution spaces. The classification aligns the courses with their didactic goals and supports students and instructors in achieving them. We systematically analyze the teaching and learning conditions of each class, such as the required effort and prior knowledge of students and instructors. Our analyses show that shared tasks are a promising teaching method for computer science education that can be adapted to different environments. However, the diversity of shared tasks also requires customized recommendations for instructors.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**; • **Applied computing** → *Education*; • **Computing methodologies** → Control methods; Natural language processing; • **Information systems** → Information retrieval; • **Theory of computation** → Machine learning theory.

This work is licensed under a Creative Commons Attribution International 4.0 License.

ITiCSE 2024, July 8–10, 2024, Milan, Italy
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0600-4/24/07
https://doi.org/10.1145/3649217.3653559

KEYWORDS

shared tasks, shared-task-based teaching, teaching design

ACM Reference Format:
Theresa Elstner, Bärbel Hanle, Frank Loebe, Maik Fröbe, Nikolay Kolyada, Janis Mohr, Jörg Frochte, Sven Hofmann, Benno Stein, and Martin Potthast. 2024. Classification of Shared Tasks Used in Teaching. In *Proceedings of the 2024 Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2024), July 8–10, 2024, Milan, Italy*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3649217.3653559

1 INTRODUCTION

Shared tasks have their origins in scientific workshops in computer science, where researchers are asked to submit working solutions for a given task, which are comparatively evaluated by the organizers in a blinded experiment. Meanwhile, shared tasks are common in many fields of computer science, especially to tackle problems without a clear-cut solution. Shared tasks are typically hosted at conferences (e.g., SemEval for natural language processing, TREC for information retrieval, NeuRIPS for machine learning, etc.)[1] where organizers publish a task's problem statement with suitable datasets as well as instructions on how to submit solutions.

Although many—including us—have been using shared tasks in computer science education for decades, their use has only recently been formalized [5]. Shared tasks specialize broader methods such as project-based learning [12] and competition-based learning [4] under the name *shared-task-based teaching*. However, our initial

[1] semeval.github.io; trec.nist.gov; neurips.cc/Conferences/2023/CompetitionTrack

formalization is limited in recognizing the diversity of shared tasks in topic and in their classroom use, and by assuming that any shared task can be aligned with a course. For example, a shared task may be inappropriate because it requires extensive prior knowledge, the timing of its associated event is not right, it cannot be replicated on site, or it does not meet formal teaching requirements.

This paper contributes to addressing these limitations. In a cross-university team of various computer science disciplines, we experimented with adapting the principles of shared-task-based teaching to different environments and requirements. Our contributions are as follows: (1) Formalization of roles and entities in a coherent terminology (Section 3.1). (2) Identification of four classes of shared tasks that frequently occur in teaching contexts (Section 3.2). Each class is analyzed according to its solution space, required effort, and organizational aspects regarding the involved stakeholders (instructors, students, shared task organizers). (3) Experience report and evaluation results of courses covering all four classes (Section 4), and recommendations for instructors (Section 5).

2 RELATED WORK

Shared-Task-Based Teaching. We build on and extend our earlier work [5], which provides the first formalization of shared-task-based teaching and lays the foundation for this new research direction. It reflects much of our own teaching experience—and we believe it applies to that of many others as well—while we apply its core ideas in a more general way. In Section 3 we add a novel perspective by considering the participants and the diversity of shared tasks, which in turn motivates us to investigate classes of shared tasks, preferably without sacrificing practicality. To our knowledge, no other work has discussed a classification of the use of shared tasks in computer science education.

Positioning Among Established Methods. As we are not aware of any other work that deals specifically with shared-task-based teaching, we briefly place this approach among more established ones. Shared task-based teaching can be associated primarily with active learning approaches [2] (which itself is grounded in constructivist learning theory [3]), especially when viewed as a particular form of project-based learning (PBL) in the sense of Bender [1]. We note that this offers overwhelming potential for comparison with numerous manifestations, specializations, and variations of PBL. Similarly, competition-based learning (CBL) is another well-known teaching/learning method that has led to many applications in computer science education and related literature (including field reports [12]). Characterized as "a method in which learning is achieved through competition" [4, p. 567], to us, shared-task-based teaching represents a blend of collaborative (within teams) and competitive (between teams) aspects for students. As such, it lies at the intersection of PBL and CBL (although approaches that integrate both, such as that of Issa et al. [8], form a subfield of their own). However, shared-task-based teaching is significantly more specific than this intersection, as it deals, for example, with working on well-defined problems, requires a clear, typically fully or semi-automated method for evaluating submitted solutions, and compares solutions over time and possibly across different courses.

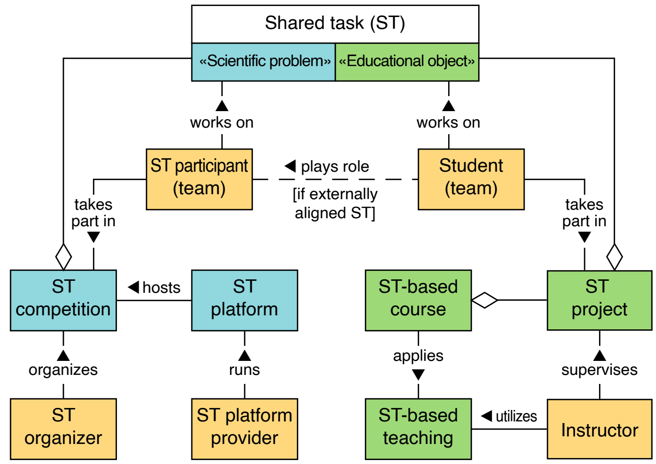

Figure 1: The two aspects of shared tasks due to the duality between science (left, blue) and education (right, green). Stakeholder roles in both domains are highlighted in yellow.

3 CLASSES OF SHARED TASKS IN TEACHING

Before classifying shared tasks in Section 3.2, the method of shared-task-based teaching is recapitulated and supplemented by a new perspective.

3.1 Shared-Task-Based Teaching

Initially, we introduced teaching with shared tasks from an instructor-driven and a process-oriented point of view [5]. The respective process model [5, Fig. 1] depicts a semester-long course aligned with a scientific shared task event. This model defines synchronization points and information objects, such as training and test data, student-produced software code, and a paper describing the students' submission. A corresponding course layout comprises five phases: (1) initial instruction, (2) analysis and initial coding, (3) student interaction through presentations, (4) final refinement and documentation of the coded solutions, and (5) final presentations and report writing. At the heart of this concept is the creation or modification of software, which is submitted as the shared task solution and which undergoes semi-automated evaluation.

Our initial work [5] focuses on the guidance of instructors by the logical–temporal structure of the method, while a stakeholder- and entity-based view is missing. In order to close this gap for the classification of variants of our method, we have developed a representation of shared-task-related notions in Figures 1 and 2. This further enables precise terminology. The notation used for all figures in this paper follows the Unified Modeling Language (UML) [14], with minor adjustments for better readability.

Figure 1 shows the concept of a *Shared Task (ST)*, its connection to the duality of science (in blue) and education (in green), and key roles (in yellow) and entities related to both of these domains. Actually, the term 'shared task' has three meanings, namely (1) a scientific competition (*ST competition*) on typically multiple (2) unsolved research problems (*«Scientific problem»*), which may yield (3) a task in an educational setting (*«Educational object»*). The topmost box *Shared Task (ST)* in Figure 1 accounts for both meanings,

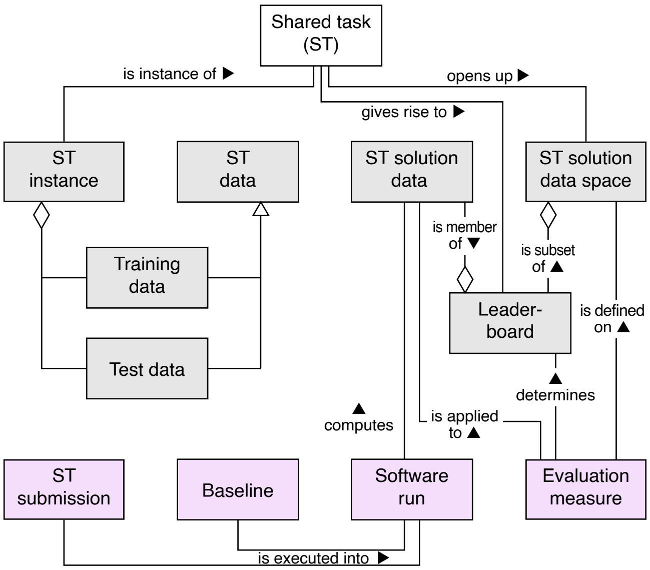

Figure 2: Information (gray) and software (purple) artifacts related to shared tasks, their solutions and solution spaces.

(2) and (3), as aspects of a task. The left-hand side of the figure (blue) includes meaning (1), i.e., *ST competition*, and names the roles of participants and organizers. These competitions are often hosted on dedicated software platforms (*ST platform*), run by providers.

As an example, we anticipate from Section 4.1 the ST competition 'SemEval 2023' [10], which included the *«Scientific problem»* 'Clickbait Spoiling' [7] in category 'Discourse and Argumentation'. It was hosted on TIRA [11], a platform designed for running blinded and reproducible computational experiments (i.e., shared tasks).

The largely green right-hand side of Figure 1 contains *ST-based teaching* at the bottom, which is utilized by *Instructors* through defining and supervising one or more *ST projects* in an *ST-based course*. *Students*, typically grouped into *teams*, take part in an ST project by solving the given shared task (similarly to *ST participants* on the scientific side). Section 4.1 reports on three ST-based courses that all used the clickbait spoiling task on the TIRA platform as *«Educational object»* of their own ST project.

Figure 2 covers very different kinds of entities related to shared tasks, namely information (in gray) and software artifacts (in purple). For running competitions, a conceptual shared task (at the top) evolves into an *ST instance*, represented by suitable *ST data* (usually *Training* and *Test data*, among other components). Solving an ST instance means to develop a software *ST submission* (bottom-left of Figure 2). Occasionally, *Baseline* software is provided for comparing results. *Software runs* of either, computed typically on an ST platform, result in *ST solution data* wrt. the shared task's input data. To compare the quality of submissions, their *ST solution data* are evaluated and ranked using one or more *Evaluation measures*. This gives rise to a *Leaderboard* and positions solutions within the *ST solution data space*. Concluding our example, the clickbait spoiling ST instance is described on a web page,[2] linking to its ST data,

[2]pan.webis.de/semeval23/pan23-web/clickbait-challenge.html

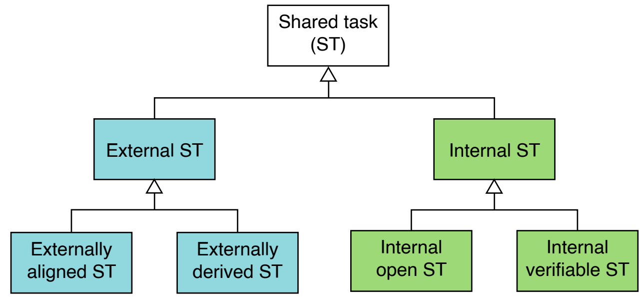

Figure 3: Practically oriented classes of shared tasks, correlated with a scientific (blue) or an educational (green) focus.

to the Baselines provided, and to its Leaderboard on TIRA.[3] Fröbe et al. [7] refer to evaluation measures such as 'balanced accuracy' in their survey of the task and its submissions.

3.2 Classes Arising from Solution Space Types

There is a wide variety of classification criteria for shared tasks. Examples include area of research, type of provided artifacts, or intrinsic task characteristics such as admitted working time, problem difficulty, software stack complexity, as well as organizational constraints (on student teams, scheduling, role responsibilities, etc.).

We propose to adopt the characteristics of solution spaces[4] as a guiding principle for our classification. Solution spaces are characterized by (1) their determinacy (open vs. closed; clarity), (2) the solution quality (what is a good/accepted solution) and rankings between solutions, and (3) the complexity of problem solving. In our experience, the resulting classes—depicted in Figure 3—correlate well with other typical criteria of shared tasks as such and of ST projects. Section 4 illustrates each class in connection with our experience reports.

Class 1: Externally Aligned Shared Tasks. These tasks run in parallel as scientific competitions at conferences such as SemEval [10]; their research problems have no satisfying solution yet (which may not exist at all), but often official baselines that must be exceeded. It is usually difficult to assess solutions, even to select or newly devise appropriate evaluation measures. The datasets are sophisticated and the technology stack for solutions tends to be demanding. Overall, the solution space of Class 1 tasks is not only open, but partially or fully unexplored, e.g., it is unknown whether and how solutions can be compared, or whether upper or lower bounds can be stated. If applicable, leaderboards reflect the state of the art.

Developing solutions in an educational setting requires considerable time. Thus, typically a single shared task is adopted as a semester project per student team. Course and competition have to be aligned, so that student submissions can yield external feedback from the shared task event. Yet instructors are free to define how they assess and evaluate learning performance.

[3]zenodo.org/record/6362726#.YsbdSTVBzrk; tira.io/task-overview/clickbait-spoiling
[4]Together with 'solution' understood more abstractly than related elements in Figure 2. For instance, the task "*Sort a given dataset according to a given sort key.*" has all conceivable implementations of sorting algorithms as its solution space.

Table 1: Classification of shared tasks and feature profiles

Shared task class			Features			
No.	Source	Mode	Task	Solution space	Evaluation measure	Complexity
1	extern.	aligned	open	unexpl.	novel	research
2	extern.	derived	open	part. expl.	est. / novel	near-res.
3	intern.	open	open	explored	established	advanced
4	intern.	verifiable	closed	known	verification	basic

Class 2: Externally Derived Shared Tasks. Shared tasks in this class draw on competitions by reusing tasks (thus being time-decoupled). This approach places more emphasis on pedagogical aspects and learning objectives. This justifies deviations (to a reasonable degree) from reused competition tasks and from outperforming baselines, e.g., to adjust the difficulty of the problem solving involved. The solution space of Class 2 tasks should generally be better known and easier to estimate than for externally aligned shared tasks. At the very least, reusing tasks comes with the benefit of pre-established evaluation measures and/or leaderboards.

Nevertheless, due to the educational focus, the assessment of student performance may also rely on or even favor individual solutions and research, based on students' innovation skills, effort, and comprehension level of the task matter. There may thus be good solutions at the bottom of a leaderboard without compromising learning success.

Class 3: Internal Open Shared Tasks. While external shared tasks may still provide inspiration or components, Class 3 shared tasks are open-ended tasks that are overall created newly and specifically for teaching purposes, driven by clear learning objectives. Instructors typically have a comprehensive understanding of the solution space and its structure. They should be aware of existing solution rankings and use established evaluation standards. Typically, there is a basic benchmark solution and a minimum quality threshold for submissions, and the latter are directly compared via a leaderboard.

This class reinforces the competitive nature of shared-task-based education. Utilizing a shared task platform is appropriate. Internal open shared tasks are usually advanced studies, but not cutting edge science. They require simpler technology than external shared tasks and allow for multiple task assignments in a project or course.

Class 4: Internal Verifiable Shared Tasks. Shared tasks of Class 4 expand the concept of shared tasks in [5] to commonly closed tasks with a fixed solution space, for which implemented solutions are submitted. This class covers typical course assignments, such as mathematical problems that require a correct solution using standard methods. The solutions are evaluated according to their correctness, with possible gradations of incorrectness. If multiple correct solutions exist, a ranking may be based on size or other factors such as submission time.

Challenging tasks with clear but unknown solutions are possible, but most Class 4 tasks are small and aim at basic, sometimes advanced content. Distant from ST projects, this class is well suitable for multiple assignments in a single shared-task-based course.

Table 2: Overview of shared task courses

Class	Institutions[a]	Subject	Year: cohort		Level[b]
1	CL, DD, RB	Natural Lang. Proc.	2023: 20		U/G
3	WE	Natural Lang. Proc.	2021: 28	2022: 40	G
2, 3	WE	Natural Lang. Proc.	2023: 62		G
2	BO	Machine Learning	2022: 27	2023: 9	U
3	BO	Robotics	2021: 29	2022: 9	U
4	BO	Control Engineering	2022: 42	2023: 33	U

[a] BO: Bochum Univ. of Applied Sciences, CL: Cologne Univ. of Applied Sciences, DD: Dresden Univ. of Technology, RB: Univ. of Regensburg, WE: Bauhaus-Univ. Weimar
[b] G: graduate, U: undergraduate

Interrelations. Table 1 positions the features initially listed for solution spaces and thus facilitates the direct comparison of the classes presented. Although they do not subsume each other, there are correlations between the class columns and their features. In particular, along the order from Class 1 to Class 4, determinacy evolves from open to closed tasks with a growing understanding of solution spaces. The ability to evaluate and compare the quality of solutions increases likewise, whereas the complexity of the actual tasks decreases from research problems to basic tasks. These observations fit well with the use of solution spaces, which we believe has led us to a useful and practically feasible shared task classification.

4 TEACHING WITH SHARED TASKS

In this section, we report on our experiences with shared tasks in university courses and we detail their respective task classes. Table 2 provides an overview of the courses.

4.1 Externally Aligned Shared Tasks

Natural Language Processing (NLP) at SemEval. Using the shared task platform TIRA [11], we organized the clickbait spoiling task [7] at SemEval 2023 [10]. Clickbait refers to online content, for example headlines, crafted to grab attention and entice clicks by appealing to curiosity. For this task, participants had to create short texts that satisfy the curiosity induced by a clickbait post.

The shared task was highly competitive and attracted a broad and diverse community with 83 registered teams from 24 countries. We accompanied a total of eight student teams from three universities. For all three courses, we held a kick-off lecture to introduce the students to the task, clarify organizational points, and create an open atmosphere that encouraged students to approach us without hesitation if they needed help or had questions.

Originally, we collected questionnaire data from all courses; however, due to a low response rate in one course, we focused on two of them from Dresden University of Technology and Cologne University of Applied Sciences. The course in Dresden had a diverse audience of undergraduate and graduate students from different majors with different levels of programming knowledge and experience in NLP, and with varying expected weekly workloads. The students worked in groups of five and had to give three presentations on their ideas, progress, and results. Each week, they could receive feedback and guidance from the instructors. In contrast, in Cologne, the course was part of a Master's degree program and was primarily attended by data science and computer science students who already had experience and basic knowledge in this field. There, the didactic approach included weekly lectures on relevant NLP topics and a consultation hour.

The feedback from all three instructors was positive overall. Compared to courses with projects without an external shared task, they found that the external shared task reduced their workload. Students responded well to the task and described it as engaging and entertaining. One instructor explicitly pointed out that, in their experience, real competition against external competitors is more motivating for the students. In the following term, the Dresden instructors supervised similar projects by students without an external shared task. They observed an equally high level of student motivation, but had to significantly reduce the number of participants due to a higher supervision load. In addition, the instructors appreciated that the organizers of the shared tasks actively communicated with the students and helped to solve technical difficulties, allowing the instructors to focus on conceptual feedback.

Wrap-up. Class 1 tasks offer students the opportunity to actively contribute to scientific progress, and instructors the convenience of a predefined task and dataset, but they require significant prior knowledge and experience on the part of the students.

4.2 Externally Derived Shared Tasks

Machine Learning (ML). Using TIRA as a shared task platform, three tasks derived from external shared tasks and freely available datasets were offered during the term. The openness of both the task and the solution methodology encouraged creative exploration. The participating students were in their sixth semester of undergraduate mechatronics or computer science and had to complete basic programming and mathematics courses beforehand.

Participants had one week to submit their solutions in the form of source code, which were then evaluated on a leaderboard using metrics such as accuracy and F1 score. The students had to beat a baseline. The choice of derived shared tasks aimed to create a controlled environment and prevented students with limited knowledge in the field from competing directly with senior ML professionals who have extensive experience. To encourage active participation and performance improvement, students were awarded bonus points based on their ranking on the leaderboard, and the quality of their submitted source code. This fostered a collaborative and inclusive learning environment, ensured fair competition, and recognition of the different skills among students.

Introduction to Natural Language Processing. In the 2023 NLP graduate course, an existing scientific shared task on the TIRA platform was utilized as the setting for a new shared task for the course's final project, with an estimated duration of 4 to 6 weeks. No restrictions were imposed with regard to the implementation approach or technology.

In the "Causal Relation Extraction" task, participants were asked to recognize whether sentences contain causal relationships (cause–effect pairs) and extract them. Each causal relation in the submitted solution was to be annotated according to a given scheme. Students were instructed to develop a program capable of identifying all causal relations in a sentence, writing their predictions to a new file that matched the format of the training dataset. Solutions had to be submitted to the shared task platform where they were ranked on a leaderboard according to predetermined metrics.

Wrap-up. Class 2 tasks require less organizational effort for the instructor, as the artifacts of the shared tasks can be reused. At the same time, students are offered a learning environment that is very similar to that of scientific shared tasks.

4.3 Internal Open Shared Tasks

Introduction to Natural Language Processing. This course, designed for graduate computer science students, involved a series of small, bi-weekly tasks to be submitted on the TIRA shared task platform. The details of the solutions were discussed in the lab classes, and additional material was provided to guide students in their implementation. Evaluations were based on automatically calculated metrics related to the problem.

Five tasks were set in the 2022 NLP course: Language Identification, Authorship Verification, Part-of-Speech Tagging, Dependency Parsing, and Lexical Substitution. For each task, the instructors provided training and validation datasets, the baselines, and solution source code templates with the auxiliary methods for data loading and evaluation. Students had to implement missing solution's key methods with restrictions on the use of "off-the-shelf" third-party libraries and generate a solution file with predictions for all given samples in the test dataset.

Besides the final project described in Section 4.2, the 2023 NLP course consisted of a series of similar smaller shared tasks based on previous years' content. They served to introduce students to the fundamentals and to acquire knowledge for the final project.

Fundamentals of Robotics. In this course, students worked together to implement a path planning algorithm using the WeBots simulation environment [9], supplemented by basic API familiarization code and introductory exercises. In this eight-week task, pairs of students worked on the complex task and submitted source code that was evaluated in both standard and dynamic obstacle simulations, focusing on efficiency, code complexity, and solution quality. Accompanying lectures on the theory of path planning and algorithms were designed to promote understanding and commitment. The task was issued close to the end of this course for undergraduate engineering students. The practical, team-based approach was praised by the instructor and the students for its openness and practical relevance.

Wrap-up. Since instructors have to provide the datasets and come up with the task description, Class 3 tasks are more organizationally demanding than external shared tasks. In return, they can offer a low-level entry point for students into the world of scientific shared tasks.

4.4 Internal Verifiable Shared Tasks

Control Engineering. In this introductory course, students had the task of using OpenModelica [6] to design a control system for a simulated pendulum. The main objective was to create a cascaded control loop. The task consisted of a sequence of detailed subtasks, each based on the principles of control engineering and starting with the attachment of a driver to the pendulum. The students

Table 3: Characterizations of entities/roles in shared task projects according to shared task classes

Class	Data	Students		Instructors		
	Size	Work time	Skill level	Adjustab. to Req.	Adjustab. to Goals	Setup and eval. effort
1	huge	months	advanced	diff.	constr.	low
2	huge	months	med.–adv.	med.	med.	low
3	med.	wks.–mths.	basic–adv.	easy	easy	med.–high
4	small	1-2 weeks	basic	easy	easy	low–med.

explored and applied different controllers in OpenModelica. Participants in this course were in the fourth semester of engineering undergraduate studies. The students' solutions featured slight variations of the controller parameters, reflecting individual preferences and interpretations, and thus giving each solution a certain uniqueness. The task was to be completed in four hours and the evaluation was based on the completion of the task rather than a ranking, which encouraged collaboration and self-reflection. This approach allowed students to focus on the control technique without external pressure. An external shared task for this course was not possible due to time constraints.

Wrap-up. Class 4 tasks are the most accessible to students and closest to traditional practice instruction, but often lack the creative aspect of the other classes.

5 DISCUSSION

Shared-task-based teaching can be used in many courses and variations. In what follows, we discuss merits of the four shared task classes as derived in Section 3.2—fully aware that many alternative systematic classifications are possible. For ours, Table 3 shows six selected characterizations of entities and actors in shared task projects, e.g., to support instructors in making decisions when designing a course with shared tasks. Although the table serves as an example for our reasoning, its entries reflect our prototypical assessments regarding the classes. More in-depth information can be derived by comparing typical effects of shared tasks in distinct classes. We illustrate this briefly with three of the properties of Table 3, again drawing on our teaching experience.

Skills Required. Externally aligned shared tasks aim to involve experts and require students to have advanced skills. Externally derived shared tasks share this characteristic, but can also provide guidance through prior solutions. Internal open shared tasks are flexible and offer tasks for different levels of expertise. Internal verifiable shared tasks are often used in introductory courses to ensure a basic understanding of the content.

Adjustability to Contextual Requirements. University programs usually specify general conditions, such as the teaching period of a semester. It can be a challenge to adapt externally aligned shared tasks to these requirements, while externally derived shared tasks can be seamlessly integrated into the semester structure. Internal shared tasks can easily be designed to meet all requirements.

Setup and Evaluation Effort. Many external shared tasks are already equipped with task descriptions, datasets, and evaluation procedures, which minimizes the effort for the instructors. Internal shared tasks usually require more effort to define and evaluate the tasks. However, many internal verifiable tasks offer straightforward evaluation procedures.

Further evidence is provided by the analysis of feedback on shared task courses. Drawing on our SemEval experience, we would like to highlight key strategies for shared task organizers to improve the participation of students bound by university curricula:

- Seek proactive communication: instructors and students will appreciate that there is someone who cares.
- Offer a kick-off event to break the ice and familiarize students with the task.
- Provide baseline code as a starting point for students.
- Align the shared task schedule with the academic year of the institutions that wish to participate.

Limitations. Surely, our classification is not the only conceivable way to categorize shared tasks. As discussed in Section 3.2, there are multiple criteria and perspectives for doing so. However, firstly, our classes are both practically oriented and systematically founded (with solution spaces as the guiding principle). Secondly, Tables 1–3 indicate that the classes are discriminatory enough to organize one's own ideas for a shared task. That said, and in the tradition of Rosch [13], these classes are of a prototypical nature, i.e., they are deliberately not precisely defined to ensure that they have fluid boundaries: Some shared tasks cannot be clearly categorized into a single class but fit multiple classes. The systematic treatment of such cases remains a future task.

6 CONCLUSION

Based on our experience in computer science education, we have generalized the approach of shared-task-based teaching and propose a new perspective as well as a classification of four shared task classes. These classes, ranging from scientific competitions to transformed assignments, have proven useful in various courses. Thus our analysis validates the promise of shared-task-based teaching.

Future work will first focus on fine-grained analyses of the motivational impact of shared tasks. In particular, we plan to examine the effects on acceptance and learning success in a differentiated way, including the setup of a longitudinal study, where courses offering shared tasks from each class are monitored and comparatively analyzed. We are further interested in the development of specific, even course-specific platforms to support different tasks. We are confident that this will improve the adaptability and impact of shared tasks in computer science education as they can be tailored to specific learning needs and objectives.

ACKNOWLEDGMENTS

We are particularly grateful to Anja Reusch, Julius Gonsior (both Dresden University of Technology) and Phillipp Schaer (Cologne University of Applied Sciences) for sharing their experience and supporting our research.

REFERENCES

[1] William N. Bender. 2012. *Project-Based Learning: Differentiating Instruction for the 21st Century.* Corwin, Thousand Oaks, CA, US.

[2] Charles C. Bonwell and James A. Eison. 1991. *Active Learning: Creating Excitement in the Classroom.* Number 1 in ASHE-ERIC Higher Education Report. George Washington University, Washington, DC, USA. https://eric.ed.gov/?id=ED336049

[3] John D. Bransford, Ann L. Brown, and Rodney R. Cocking (Eds.). 2000. *How People Learn: Brain, Mind, Experience, and School.* National Academy Press, Washington, D.C.

[4] Juan Carlos Burguillo. 2010. Using Game Theory and Competition-based Learning to Stimulate Student Motivation and Performance. *Computers & Education* 55, 2 (Sept. 2010), 566–575. https://doi.org/10.1016/j.compedu.2010.02.018

[5] Theresa Elstner, Frank Loebe, Yamen Ajjour, Christopher Akiki, Alexander Bondarenko, Maik Fröbe, Lukas Gienapp, Nikolay Kolyada, Janis Mohr, Stephan Sandfuchs, Matti Wiegmann, Jörg Frochte, Nicola Ferro, Sven Hofmann, Benno Stein, Matthias Hagen, and Martin Potthast. 2023. Shared Tasks as Tutorials: A Methodical Approach. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence* (Washington DC, USA, February 7-14, 2023) *(AAAI'23/IAAI'23/EAAI'23)*, Brian Williams, Yiling Chen, and Jennifer Neville (Eds.). AAAI Press, Washington DC, USA, 15807–15815. https://doi.org/10.1609/aaai.v37i13.26877

[6] Peter Fritzson, Adrian Pop, Karim Abdelhak, Adeel Asghar, Bernhard Bachmann, Willi Braun, Daniel Bouskela, Robert Braun, Lena Rogovchenko-Buffoni, Francesco Casella, Rodrigo Castro, Rüdiger Franke, Dag Fritzson, Mahder Gebremedhin, Andreas Heuermann, Bernt Lie, Alachew Mengist, Lars Mikelsons, Kannan Moudgalya, and Per Östlund. 2020. The OpenModelica Integrated Environment for Modeling, Simulation, and Model-Based Development. *Modeling, Identification and Control: A Norwegian Research Bulletin* 41, 4 (Oct. 2020), 241–295. https://doi.org/10.4173/mic.2020.4.1

[7] Maik Fröbe, Tim Gollub, Benno Stein, Matthias Hagen, and Martin Potthast. 2023. SemEval-2023 Task 5: Clickbait Spoiling, See [10], 2275–2286. https://doi.org/10.18653/v1/2023.semeval-1.312

[8] Ghassan Issa, Shakir M. Hussain, and Hussein Al-Bahadili. 2014. Competition-Based Learning: A Model for the Integration of Competitions with Project-Based Learning using Open Source LMS. *International Journal of Information and Communication Technology Education* 10, 1 (2014), 1–13. https://doi.org/10.4018/ijicte.2014010101

[9] Olivier Michel. 2004. Cyberbotics Ltd. Webots$^{\text{TM}}$: Professional Mobile Robot Simulation. *International Journal of Advanced Robotics Systems* 1, 1 (2004), 39–42. https://doi.org/10.5772/5618

[10] Atul K. Ojha, Ayşe Seza Doğruöz, Giovanni Da San Martino, Harish Tayyar Madabushi, Ritesh Kumar, and Elisa Sartori (Eds.). 2023. *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023), Toronto, Canada, July 13-14, 2023* (Toronto, Canada, July 13-14, 2023). Association for Computational Linguistics, Stroudsburg, PA, USA. https://aclanthology.org/2023.semeval-1.0

[11] Martin Potthast, Tim Gollub, Matti Wiegmann, and Benno Stein. 2019. TIRA Integrated Research Architecture. In *Information Retrieval Evaluation in a Changing World: Lessons Learned from 20 Years of CLEF*, Nicola Ferro and Carol Peters (Eds.). The Information Retrieval Series, Vol. 41. Springer, Berlin, 123–160. https://doi.org/10.1007/978-3-030-22948-1_5

[12] Robert Pucher and Martin Lehner. 2011. Project Based Learning in Computer Science: A Review of More than 500 Projects. *Procedia Social and Behavioral Sciences* 29 (2011), 1561–1566. https://doi.org/10.1016/j.sbspro.2011.11.398 The 2nd International Conference on Education and Educational Psychology (Istanbul, Turkey, October 19-22, 2011) *(ICEEPSY 2011)*.

[13] Eleanor Rosch. 1978. Principles of Categorization. In *Cognition and Categorization*, Eleanor Rosch and Barbara B. Lloyd (Eds.). Lawrence Erlbaum, Hillsdale, NJ, USA, 27–48.

[14] James Rumbaugh, Ivar Jacobson, and Grady Booch. 2005. *The Unified Modeling Language Reference Manual* (2. ed.). Addison Wesley, Reading, Massachusetts.