

MATEMATIKA И МАТЕМАТИЧЕСКО ОБРАЗОВАНИЕ, 2012
MATHEMATICS AND EDUCATION IN MATHEMATICS, 2012
*Proceedings of the Forty First Spring Conference
of the Union of Bulgarian Mathematicians
Borovetz, April 9–12, 2012*

**CORE: A MULTI-PURPOSE PROGRAMMING CONTEST
REPOSITORY SYSTEM***

Yani Chaushev, Miloslav Sredkov, Krassimir Manev

Every programming contest uses a set of software tools to manage contest processes. While these instruments usually cover the specifics of the particular contest format sufficiently, they rarely address the issues of long term storage and interoperability. In this paper, we present one software tool attempting to tackle these problems. Instead of being a complex platform concerning every aspect of a programming contest, CORE is a contest repository system providing a central place where contest data is stored and maintained. The current status of the repository along with its main elements are described, and the upcoming planned features are presented.

1. Introduction. There are a different types of programming contest and numerous grading systems (GS) that are used for their management [1]. Each GS has own software tools, different definitions of resources and different resource formats. Recently, GS are also used in the educational process to facilitate grading of students, e.g. for examinations and checking homework assignments [2]. Very often the contest systems need to be tweaked and customized in order to fit into the specifics of the process they are needed for.

Having independent systems leads to obvious problems. They have different formats and associated tools, which are very often incompatible. Transferring data from one system to another is difficult at best and requires additional programs or manual work. Overall, systems of different types stay isolated from each other. Even using only one system leads to certain problems. Because of the specifics and amount of resources, it is not trivial to gather, store and serve them properly. For example, often there is a need to work in an off-line environment, which usually results in tasks or results being lost because of the lack of an easy mechanism for merging the resources back.

There are already attempts to tackle these problems. Verhoeff [3] tried to solve the problem by proposing a specific format named *Peach Exchange Format* which strictly defines how problems and their related artifacts are stored, though the proposal is of high complexity and did not impose as popular. More recently, Manev et al. [4] outlined a software platform aiming to integrate different grading systems into a single environment for the educational process. The construction of such an environment, however, requires

*2000 Mathematics Subject Classification: 68N99.

Key words: education in programming, programming contests, grading systems.

This work was partially supported by the Research Fund of Sofia University through contract 208/2011.

significant efforts; in fact, it can be seen as a proper superset of the one currently being described.

In this paper we present the CORE system, which is designed to attack these problems. Unlike other systems that have all tools integrated into them, CORE's goal is to be simple and serve only as a repository. It defines formats for resources that are consistent with the needs of existing systems and provides means for exchanging data with other systems and integration with external tools, such as grading systems. In section 2 expected practical uses of CORE are described. Section 3 is overview of the system. In section 4 we present the resource format that is defined so far. Section 5 describes the security policy. Section 6 is about the current status and planned features of CORE, and in section 7 some conclusions are shared.

2. Practical uses. The CORE system can be the backbone of a system for programming contests or educational programming. Since its purpose is to be a repository, it can help with running a programming contest by doing the following:

- serve the problems to the appropriate people in compliance with a security policy;
- accept solutions to some restrictions;
- display the results in an appropriate format.

The one thing missing is grading the solutions. Since we want CORE to be more universal, it is natural to move the grader outside and provide means for integration with third party tools instead.

The users of CORE are varying in a wide range. We have a group of contest organizers, who will manage the life-cycles of contests, collect and organize task, create contest, share resources and archive past events. Perhaps, the largest group is the competitors who will have the opportunity to access tasks, submit solutions and see results and statistics. This can be done directly via a web interface or indirectly through a third party system. The last group of users includes judges and administrators. They can grade solutions or correct tasks and competitors' scores when necessary.

3. CORE overview. The name CORE comes for short of "Contest Repository". CORE is in development and many of the features are already implemented. The system is implemented in Java and is designed to run as an application in a web server. All of the resources and configuration are stored in the file system (i.e. no database) for the purpose of easiness of management. It has a web interface for interacting with users. Administration is done by editing configuration files or by using command line tools.

Grading a solutions is a complicated process and is not part of CORE. Instead, CORE is designed to be able to collaborate with grading systems, which takes all of the burden of running a solution and testing if it is valid. For that purpose, and for the purpose of integration with other external tools and software, CORE is planned to provide access to its functionality through an appropriate set of RESTful web services.

4. Contest resources. While CORE keeps various types of information such as user's data or submitted solutions, the most essential resources are the ones that define the programming tasks and their organization. During the design of the formats the concrete needs of the systems Maycamp Arena [5] and spoj0 [6] were taken into account, but the design aims to be general enough so that it can suite most grading systems. We are defining the following types of such resources that CORE recognizes:

- *series* – represents a top-level wrapper for a number of *contests*. For example it can correspond to a course and be named `‘‘daa_2010’’`.

- *contest* – a container of a number *tasks* corresponding to an event in the context of conducting a programming challenge like an exam, a homework or a programming contest (e.g. ‘‘homework_01’’).
- *problem* (or task) – represents a particular programming task.

All of the resources and configuration are stored in the file system. Since each resource can contain a number of assets, it is natural to design a resource as a directory that contains its assets. Its name matches the pattern `[a-z0-9][a-z0-9_-]*` and serves as a resource identifier. The following illustrates the structuring of the resources:

```
(root)
* series/
  * <series-name>/ (one per series)
    * _files/
    * _public_files/
    * <contest-name>/ (one per contest)
      * _files/
      * _public_files/
      * <problem-name>/ (one per problem)
        * _security.json
        * problem.json
        * (problem related files)
      * _security.json
      * contest.json
    * _security.json
    * series.json
  * _security.json
* souls/
  * users/
    * <user-name> (one for each user)
      * user.json
```

Each resource contains meta information in one or more files, each of them containing a JSON [7] object encoded in UTF-8. The JSON format is very simple and we will briefly describe it. There are *objects*, which are surrounded with curly brackets and have keys and values:

```
{"color": "red", "size": 8}
```

There are also *arrays* (or lists), surrounded with square brackets:

```
["red", "green", "pink"]
```

A value can be a double quoted *string*, *number* written as a decimal integer or fraction, an *object*, or one of the literals `true`, `false`, `null`.

4.1. Series format. The *series* directory contains a descriptor file named `series.json` and a sub-directory for each of the contained *contests*. The descriptor contains a format identifier, the full title of the *series* and optionally short public and private textual notes. Example:

```
{
  "format": "core-series-1",
  "title": "Design and analysis of algorithms - 2010",
  "about": "Lectors:..., Hours:...etc.",
  "notes": "(Some sensitive information)"
}
```

4.2 Contest format. The directory of each *contest* contains a descriptor named `contest.json`, and a sub-directory for each of the contained *problems*. The descriptor contains a format identifier, the full title of the *contest*, its start time, its duration, information about grading and other information. Example:

```
{
  "format": "core-contest-1",
  "title": "Homework 1",
  "start_time": "2012-06-28T09:00:00Z",
  "duration": 300,
  "about": "",
  "grading_style": "acm",
  "problem_order": ["fish", "honey", "swim"],
  "problemScores": [120, 150, 80]
}
```

4.3 Problem format. The directory of each *problem* contains a descriptor named `problem.json`, a number of files containing the problem text (e.g. `description-en.pdf`), input and answer files for the test data, and eventually files for the checker. The descriptor contains a format identifier, the full title of the problem, and additional private and public information including instructions for grading systems. Example:

```
{
  "format": "core-problem-1",
  "title": "hw1-problem-A",
  "about": "",
  "time_limit": 4.5,
  "memory_limit": 64,
  "origin": "Taken from Contest-X",
  "notes": "Dynamic Programming",
  "authors": ["author1", "author2"],
  "test_weights": [5, 10, 10, 10, 15],
  "checker": "diff"
}
```

5. Security policy. As analyzed by Tochev and Bogdanov [8], the security of programming contest system is a complicated problem. Fortunately, most of the challenges are associated with the grading process, which is not handled by CORE. Still, keeping problem descriptions, solutions and test data secret is crucial during the preparation and execution of programming contests. Since a repository's main goal is to store and serve such resources, a security policy defining who has access to the system and to what extend, as well as how these permissions can be managed, is essential.

This is achieved mainly in three directions. First, the assets of the contest resources are considered either *private* or *public*, depending on a sophisticated rules related to contest rules, current time, etc. In addition, sub-directories named `_public_files` and `_files` can contain supplementary public and private resources accordingly. Secondly, multiple types of access are defined, such as listing, viewing the public assets, and fully viewing the resource. The access permissions are recursive and can be defined in a file named `_security.json` for each resource individually. Finally, authentication using SHA256 hashing is employed in order to enforce the defined restrictions.

6. Current status and planned features. The project has many if its features already implemented: it is capable of storing resources; it has a working security policy; its web interface provides basic means to browse and manage resources. The current state of CORE allows it to be used for archiving past events.

In the near future CORE should have the option of synchronizing with other instances of the same system. A typical use-case is setting up an instance of the system in an offline environment, conducting a contest, and finally merging the results back to the main repository. Having a mechanism for synchronization makes losing information about the contest less likely.

7. Conclusion. We described in details the concept of the CORE system. We pointed out its main parts, structure and security policy. The simple and clear concept for being what it is, i.e. a repository, gives us confidence that it will soon be capable of helping with the process of organizing programming events and educational courses. When this happens, its actual usage will allow us to determine more precisely the directions in which it needs to be developed in future.

REFERENCES

- [1] KR. MANEV, M. SREDKOV, TS. BOGDANOV. Grading Systems for Competitions in Programming, *Math. and Education in Math.*, **38** (2009), 103–116.
- [2] KR. MANEV, M. SREDKOV, TS. BOGDANOV, V. MIHOV. Grading Systems in Teaching of Programming, to appear.
- [3] T. VERHOEFF. Programming task packages: Peach exchange format. *Olympiads in Informatics*, **2** (2008), 192–207.
- [4] KR. MANEV, M. SREDKOV, P. ARMYANOV. Software Plaform for Teaching Programming with Grading Systems. *Math. and Education in Math.*, **40** (2011), 300–305.
- [5] V. MIHOV. Maycamp arena project — using grading systems for education in programming. *Math. and Education in Math.*, **40** (2011), 438–443.
- [6] M. SREDKOV. Spoj0 — a system for programming contests. Master’s thesis, Sofia University, July 2007.
- [7] D. CROCKFORD. The application/json media type for JavaScript Object Notation (JSON). 2006. <http://www.ietf.org/rfc/rfc4627>
- [8] T. TOCHEV, TS. BOGDANOV. Validating the Security and Stability of the Grader for a Programming Contest System. *Olympiads in Informatics*, **4** (2010), 113–119.

Yani Chaushev, Miloslav Sredkov, Krassimir Manev
Faculty of Mathematics and Informatics
St. Kliment Ohridski University of Sofia
5, J. Bourchier Blvd
1164 Sofia, Bulgaria
e-mail: sleepwalker.bg@gmail.com
miloslav@gmail.com
manev@fmi.uni-sofia.bg

CORE — МНОГОЦЕЛЕВА СИСТЕМА-ХРАНИЛИЩЕ ЗА СЪСТЕЗАНИЯ ПО ПРОГРАМИРАНЕ

Яни Чаушев, Милослав Средков, Красимир Манев

Всяко състезание по програмиране използва множество софтуерни инструменти за управление на процесите по време на състезанието. Въпреки че тези инструменти обикновено покриват спецификите на конкретния вид състезание задоволително, те рядко адресират трудностите на дългосрочното съхранение на данни и оперативната съвместимост. В тази статия е представен един софтуерен инструмент адресиращ тези проблеми. Вместо комплексна система, касаеща всички аспекти на състезанието, CORE е централизирано хранилище за съхраняване и поддържане на необходимите за различни състезания данни. Представени са основните му елементи, текущото състояние на реализацията и перспективите за развитие на системата.