Mastery Grids for Java

Personalized Adaptive Web Systems Lab, School of Information Sciences, University of Pittsburgh
Learning+Technology research group, Aalto University, 2015

Mastery Grids for Java is a practice system for Java programming that offers students a personalized access to several kinds of smart learning content and allows them to monitor their learning progress and knowledge in comparison with pairs or groups, and allows teachers to create such smart learning contents with our authoring tools and design a course curriculum with desired topics and matching learning materials. Our past research has successfully deployed Mastery Grids in various learning domains (Java, Python, SQL) in different universities. Java programming language is one of the already-covered learning domains in Mastery Grids. The Java programming course in Mastery Grids has been used for undergraduate and graduate level course since 2013. Although the course material covers an extensive range of Java programming topics, it can be extended using the authoring tools that come along with Mastery Grids. Our research shows that Mastery Grids effectively increases learners' performance, motivation, engagement and retention.

❏ **Mastery Grids Interface Features**

- ❏ **Knowledge and Progress Visualization:** Presenting student learning progress and knowledge by colored grids from four perspectives as shown in Figure 1: students' perspectives ("Me" or "Group") and comparison perspectives.
- ❏ **Social Comparison:** Implemented by "Me vs Group" and "Me vs other learners" in the visualization in various granularities including by topics or by learning content type (Figure 1, Figure 2).
- ❏ **Adaptive Navigation Support:** Recommendation of learning resources are provided based on student activities and learner knowledge in a centralized user modeling server. (see red stars in Figure 1)

Figure 1: Mastery Grids' knowledge and progress visualization of Me, Me vs Group, Group, other learners embedding open learner model of oneself and social comparisons of a student with the group or other learners.
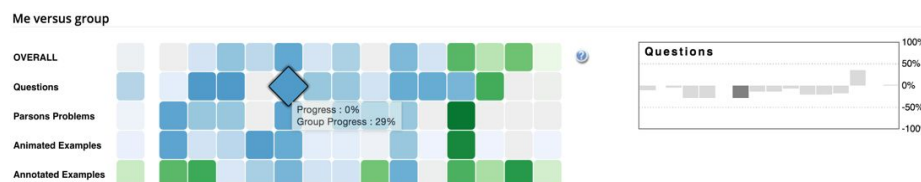

Figure 2: Mastery Grids' social comparison by learning content (resource) type.

❏ **Java Learning Smart Content and Flexible Course Design**

Mastery Grids provides multiple types of interactive, smart learning contents including parameterized Java quizzes, and annotated examples. Learning material in Java course are organized in topics. Topics are aligned in accordance with learning timeline as is represented in Figure 1.

❏ **Parameterized Semantics Problems:** Short-answer quizzes that test for students' understanding of code snippets and how they work. These questions can be repeated by the students, every time with different parameters. (Figure 3 Left)

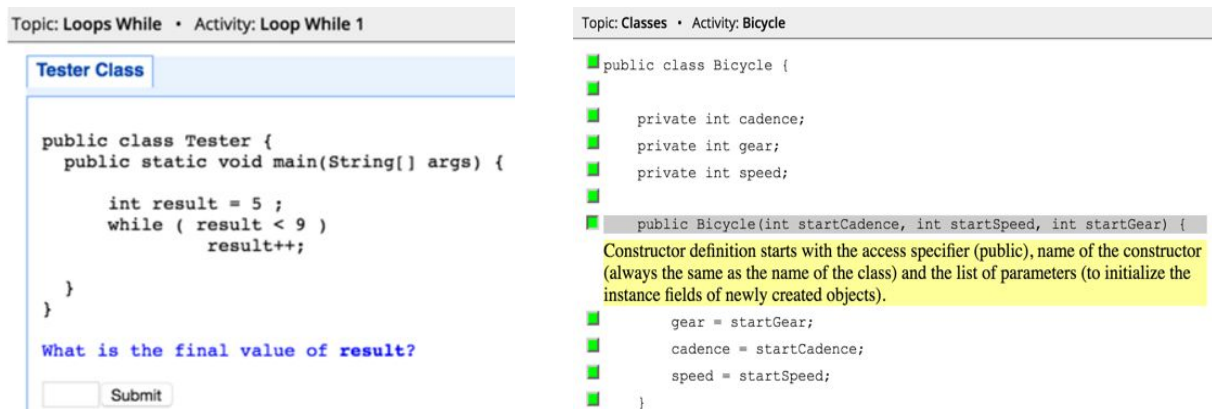❏ **Annotated Examples:** Short snippets of code with explanation of key lines in them. (Figure 3 Right)

Figure 3: The left side shows a parameterized semantics problem, and the right side shows an annotated example.

❏ **Authoring Tools:** We have developed a set of authoring tools enabling the above content creation and a complete course design using these content. Tools include

   ❏ Content Authoring for creating smart learning contents
   ❏ Course Authoring for creating adaptive courses
   ❏ Group Authoring for managing users and groups
   ❏ Authoring Portal to access different authoring tools

❏ **Learning Analytics Enabled**

Mastery Grids with its architecture provides following major advantages for conducting learning analytics:
   ❏ It allows flexibly experimenting different learner models, recommendation/personalization techniques.
   ❏ Its fine-grained logging mechanism for user activities allows for behavior analysis such as sequential pattern mining.
   ❏ It allows easily incorporating new smart content on which one can conduct experiments to study the effect.

❏ **Resources and Documentations**

Our project website with detailed information can be accessed here bit.ly/OSLM_MG. All related code can be downloaded from Github: https://github.com/PAWSLabUniversityOfPittsburgh, https://github.com/acos-server/.