

HỆ THỐNG TỰ ĐỘNG CHẤM BÀI TẬP LẬP TRÌNH

AN AUTOMATIC ASSESSMENT SYSTEM FOR PROGRAMMING EXERCISES

Trần Thị Mỹ Tiên

Trường Đại học Giao thông vận tải Thành phố Hồ Chí Minh

Tóm tắt: Ngày nay, tự động hóa đang là xu hướng chính trong nền công nghiệp 4.0, tại khoa Công nghệ thông tin Trường Đại học Giao thông vận tải Thành phố Hồ Chí Minh, việc chấm bài tập lập trình của sinh viên đang được thực hiện một cách thủ công. Do đó giảng viên mất rất nhiều thời gian và công sức cho việc chấm bài. Ngoài ra, sinh viên phải chờ đợi đến lượt để được giảng viên chấm bài. Việc chấm bài cũng chỉ được thực hiện trên lớp với khoảng thời gian giới hạn. Bài báo này đề xuất một hệ thống phần mềm có thể chấm bài tập lập trình của sinh viên một cách tự động. Hệ thống sẽ giúp giảng viên tiết kiệm thời gian và công sức cho việc chấm bài. Hệ thống hoạt động trên nền tảng web nên sinh viên có thể sử dụng hệ thống mọi lúc, mọi nơi. Qua đó, bài báo cũng trình bày kết quả đạt được sau một thời gian triển khai hệ thống.

Từ khóa: Hệ thống chấm bài tự động, bài tập lập trình, chấm bài trực tuyến.

Chỉ số phân loại: 1.4

Abstract: Nowadays, automation is main trend in industry 4.0, at Information Technology faculty, Ho Chi Minh city University of Transport, assessment of programming exercises is being done manually. Therefore, it takes a lot of time and efforts for lecturers to assess students' assignments. Besides, students have to wait for their turn to be assessed. Assessment is only be handled at classroom with a short time. This paper proposes a system that can assess students' assignments automatically. The system will help lecturers save their time and effort. The system is a web-based platform, so students can use it anytime, anywhere. This paper also presents the results achieved by developing and testing the system.

Keywords: Automatic assessment system, programming exercises, online assessment

Classification number: 1.4

1. Giới thiệu

Hệ thống có thể tự động chấm bài tập lập trình của sinh viên đã được các trường đại học trên thế giới thiết kế và ứng dụng trong nhiều năm qua [1]. Đặc biệt trong những năm gần đây, các hệ thống tự động chấm bài tập lập trình được phát triển và ứng dụng ngày càng nhiều hơn trong lĩnh vực giáo dục và trong cả các lĩnh vực khác như tuyển dụng, việc làm [2][3].

Hiện nay, ACM-ICPC được xem là kỳ thi lập trình lâu đời và danh giá nhất dành cho sinh viên các trường đại học và cao đẳng trên toàn cầu. Trong những năm đầu tiên tổ chức kỳ thi, hệ thống chấm bài bán tự động đã được ứng dụng để chấm điểm các bài tập lập trình của các đội thi. Qua thời gian, hệ thống chấm bài được nâng cấp theo hướng tự động hóa ngày càng cao [5]. Ở Việt Nam, kỳ thi Olympic Tin học sinh viên toàn quốc được tổ chức đều đặn hàng năm. Kỳ thi được tổ chức

lần đầu vào năm 1992. Lúc đó, các bài thi lập trình cá nhân được chấm bằng hình thức bán tự động. Từ năm 2016 đến nay, ban tổ chức đã triển khai hệ thống chấm bài tập lập trình tự động hoàn toàn cho các khối thi cá nhân.

Dựa trên sự phát triển mạnh mẽ của nền tảng Internet, các khóa học lập trình trực tuyến được mở ra ngày càng nhiều. Người học có thể tham gia học tại mọi thời điểm, mọi nơi phù hợp với điều kiện của cá nhân mình. Trong quá trình học tập trực tuyến, các hệ thống chấm bài tự động cũng sẽ được sử dụng như một lễ tất yếu.

Ngoài ra, các hệ thống đánh giá tự động đã và đang được ứng dụng ngày càng nhiều vào quá trình tuyển dụng, đánh giá lập trình viên. Với phiên bản thử nghiệm ban đầu, hệ thống Codility đã đem lại hiệu quả cao trong việc hỗ trợ công ty sàng lọc ứng viên tham gia tuyển dụng vào vị trí lập trình viên. Trong vòng sơ loại, công ty đưa ra một số bài

toán và yêu cầu tất cả ứng viên làm bài qua hệ thống chấm tự động. Hơn 90% ứng viên bị loại do không làm đúng trọn vẹn các bài toán dễ nhất trong số đó [6].

Bài báo này đề xuất một hệ thống trực tuyến, có thể tự động chấm bài lập trình của sinh viên. Qua đó, góp phần tăng hiệu quả của việc dạy và học các môn lập trình. Hai tác nhân tương tác chính với hệ thống là giảng viên và sinh viên. Giảng viên sẽ ra các bài tập và xem kết quả làm bài của sinh viên.

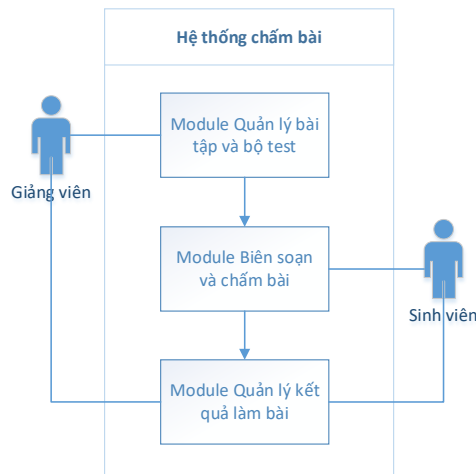
Sinh viên có thể làm bài bằng trình biên soạn mã nguồn được tích hợp ngay trên hệ thống hoặc bằng một công cụ biên soạn mã nguồn khác. Sau khi làm bài xong, sinh viên nộp bài cho hệ thống. Hệ thống sẽ tự động chấm bài bằng cách nhập vào lần lượt từng bộ test và đối chiếu kết quả đầu ra. Sau đó, hệ thống hiển thị kết quả tổng hợp của bài làm. Mô hình tương tác của hệ thống được minh họa theo hình 1.



Hình 1. Mô hình tương tác của hệ thống tự động chấm bài lập trình.

2. Kiến trúc hệ thống

Hệ thống gồm ba module chính được minh họa qua hình 2.



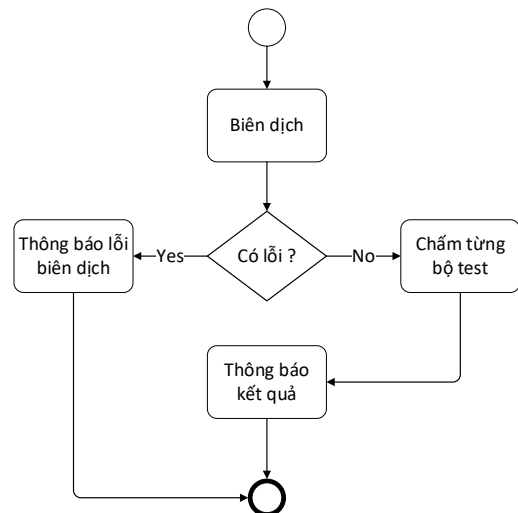
Hình 2. Kiến trúc hệ thống.

2.1. Module Quản lý bài tập và bộ test

Bao gồm các chức năng cho phép giảng viên có thể đưa bài tập lên hệ thống. Bài tập được chia thành nhiều chủ đề với nhiều mức độ khác nhau. Để đánh giá mức độ tối ưu của bài lập trình, giảng viên sẽ quy định thêm thời gian chạy tối đa cho nó. Mỗi bài tập đưa lên sẽ kèm theo một hoặc nhiều bộ test để đánh giá bài làm của sinh viên. Mỗi bộ test bao gồm dữ liệu đầu vào (input) và đáp án kết quả đầu ra (output).

2.2. Module Biên soạn và chấm bài

Bao gồm các chức năng cho phép sinh viên trực tiếp biên soạn mã nguồn bằng nhiều ngôn ngữ lập trình khác nhau gồm C/C++, Java, Pascal. Chức năng chấm bài bắt đầu khi hệ thống nhận được bài làm của sinh viên. Quy trình chấm bài được minh họa bằng lưu đồ trong hình 3.



Hình 3. Lưu đồ quy trình xử lý chấm bài.

Đầu tiên, hệ thống sẽ tiến hành biên dịch bài làm. Nếu biên dịch có lỗi, hệ thống sẽ thông báo chi tiết lỗi để sinh viên biết và sửa. Lỗi biên dịch thường là những sai sót về chính tả, cú pháp. Nếu biên dịch thành công, hệ thống sẽ lần lượt lấy các bộ test của bài

tập để chấm. Với mỗi bộ test, hệ thống sẽ chạy chương trình và đưa input vào, sau đó chờ nhận output của chương trình. Nếu chương trình chạy quá thời gian tối đa cho phép thì hệ thống sẽ ngắt chương trình và xem như sinh viên chưa làm đúng bộ test này. Khi có output từ chương trình, hệ thống sẽ so sánh với đáp án để biết được kết quả đúng hay sai. Cuối cùng, hệ thống sẽ tổng hợp và thông báo kết quả bài làm cho sinh viên.

2.3. Module Quản lý kết quả bài làm

Bao gồm các chức năng cho phép sinh viên xem lại kết quả làm bài của mình. Sinh viên có thể mở lại những bài đã làm, chỉnh sửa và nộp để hệ thống chấm lại. Giảng viên cũng có thể xem kết quả làm bài của sinh viên.

3. Các vấn đề cần quan tâm thêm

Phần này đề cập đến một số vấn đề xảy ra trong thực tế cần quan tâm thêm khi xây dựng và triển khai hệ thống.

3.1. Bảo mật

Vì bài làm của sinh viên được nhập vào hệ thống là những đoạn mã chương trình nên vấn đề bảo mật cần phải được chú ý đặc biệt. Vì nếu thực thi một đoạn mã virus hoặc một đoạn mã độc thì máy chủ sẽ bị nhiễm virus hoặc bị chiếm quyền điều khiển. Ngoài ra, một số lỗi trong bài lập trình của sinh viên cũng có thể gây ra ảnh hưởng lớn cho toàn hệ thống. Có hai hướng tiếp cận chính cho vấn đề này đó là giải quyết nó hoặc chấp nhận sự tồn tại của nó.

Hướng tiếp cận thứ nhất là giải quyết vấn đề. Có nhiều phương pháp để giải quyết vấn đề này. Một trong những phương pháp được sử dụng phổ biến là dùng cơ chế sandbox. Hệ thống của Cerioli và Cinelli [6] thiết lập môi trường được cô lập và giới hạn quyền tối thiểu để thực thi các đoạn mã của cả giảng viên và sinh viên. Một phương pháp khác là dùng biểu thức chính quy (Regular Expression) để xây dựng bộ lọc loại bỏ những đoạn mã độc [2]. Hoặc đơn giản hơn là dùng một phần mềm diệt virus hiện có để quét tập tin chương trình trước khi thực thi.

Hướng tiếp cận thứ hai là chấp nhận sự tồn tại của vấn đề. Theo kinh nghiệm của

Pieterse [4], các lỗi nếu có trong các bài lập trình của sinh viên là do các sai sót thuần túy hơn là các mã độc có chủ ý. Theo báo cáo của Chen [7], trong 5 năm hoạt động, hệ thống không nhận được bất cứ đoạn mã độc nào từ sinh viên gửi lên. Có thể do sinh viên nghiêm túc hoặc đơn giản là sinh viên hiểu được hậu quả của hành động gửi một mã độc lên hệ thống.

Hệ thống hiện tại đang lựa chọn cách tiếp cận thứ nhất để giải quyết vấn đề bảo mật này. Tác giả thiết lập một user trên máy chủ, user này không có quyền gì khác ngoài quyền thực thi bài làm của sinh viên. Nếu bài làm là mã độc thì cũng sẽ không can thiệp được vào hệ thống.

3.2. Đạo văn

Khi sử dụng một hệ thống tự động để đánh giá kết quả học tập của sinh viên. Việc sinh viên đạo văn hay sao chép bài của sinh viên khác có khuynh hướng tăng cao hơn so với việc chấm thủ công [4]. Để hạn chế được vấn đề này, giảng viên nên kết hợp việc nhắc nhở sinh viên về thái độ và tinh thần tự trọng trong học tập cùng với việc xử lý nghiêm khắc những sinh viên vi phạm. Hệ thống có thể được tích hợp thêm một tiện ích nhằm hỗ trợ giảng viên phát hiện đạo văn. Ngoài ra, hệ thống có thể được kết hợp với một số hệ thống phát hiện đạo văn hiện có. Khi sinh viên gửi bài cho hệ thống, hệ thống sẽ chuyển bài làm của sinh viên tới hệ thống phát hiện đạo văn, hệ thống đạo văn sẽ trả về kết quả phân tích bài làm hiện tại, nếu tỉ lệ sao chép (%) vượt ngưỡng cho phép sẽ thông báo đến sinh viên và cho điểm 0 bài làm này.

4. Kết quả thử nghiệm

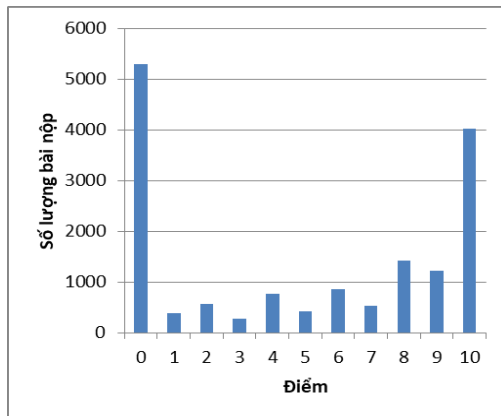
Dựa vào mô hình đề xuất, tác giả đã phát triển một hệ thống chấm bài tự động (<http://coding.ut.edu.vn:1010/>). Hệ thống đã được triển khai cho các lớp học lập trình của khoa Công nghệ thông tin, Trường Đại học Giao thông vận tải Thành phố Hồ Chí Minh và các sinh viên tham gia ôn luyện Olympic Tin học của Trường từ tháng 9/2016 đến tháng 5/2019.

Hệ thống được xây dựng trên nền tảng web, bằng ngôn ngữ PHP và hệ quản trị cơ sở dữ liệu MySQL. Số bài tập đưa vào hệ

thống là 81, tổng số bộ test là 594, số sinh viên tham gia thử nghiệm là 914.

Tổng số lần nộp bài của sinh viên là 15.734. Đây là tổng số bài nộp không bị lỗi biên dịch. Những bài làm có lỗi biên dịch thì hệ thống không lưu trữ lại mà chỉ thông báo lỗi cho sinh viên.

Thông tin thống kê kết quả chấm bài được thể hiện qua hình 4. Số bài làm bị điểm 0 chiếm số lượng nhiều nhất. Khi rà soát lại những bài làm này, tác giả nhận thấy nguyên nhân chính là do lỗi nhập xuất. Sinh viên vẫn còn quen với kiểu nhập xuất thủ công nên không làm đúng yêu cầu nhập xuất theo kiểu tự động của hệ thống.

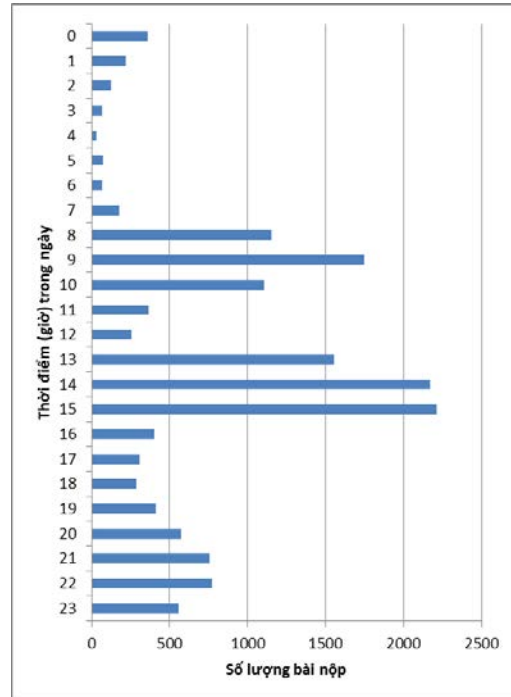


Hình 4. Biểu đồ tổng hợp kết quả chấm bài.

Thông tin thống kê thời điểm sinh viên làm bài được thể hiện qua hình 5. Khoảng 2/3 số lượng bài nộp vào khung thời gian học tập trên lớp, tập trung nhiều nhất từ 8 đến 10 giờ sáng và từ 13 đến 15 giờ chiều. Ngoài thời gian học tập trên lớp, sinh viên tập trung làm bài nhiều nhất vào thời điểm từ 21 đến 22 giờ tối.

Kết quả thử nghiệm hệ thống cho các lớp học lập trình thể hiện rất rõ tính hiệu quả của hệ thống. Giả sử, mỗi bài gồm 10 bộ test, mỗi bộ test giảng viên mất 30 giây để nhập và kiểm tra kết quả thủ công. Đối với các bài tập phức tạp và cần kiểm tra với bộ dữ liệu lớn, thời gian có thể tăng lên nhiều. Trong khi đó, máy chấm chỉ mất chưa đầy 1 giây cho việc nhập và so khớp 1 bộ test. Như vậy nếu chấm tự động thì thời gian chấm giảm 30 lần. Ngoài ra, hệ thống hoạt động 24/7, sinh viên có thể nộp bài mọi lúc và nhận kết quả nhanh chóng tức thì. Ngược lại, với cách chấm thủ công, nếu chấm ngay tại lớp sinh

viên phải chờ tới lượt hoặc chờ nhận kết quả rất lâu sau đó nếu giảng viên chấm ngoài giờ lên lớp. Vậy khi áp dụng hệ thống cho việc giảng dạy lập trình, giảng viên tiết kiệm được rất nhiều thời gian chấm bài. Đồng thời, hệ thống sẽ là công cụ hữu hiệu giúp sinh viên tự rèn luyện thêm ngoài giờ lên lớp.



Hình 5. Biểu đồ thống kê thời điểm làm bài.

Một ưu điểm khác của hệ thống là đánh giá mức độ tối ưu về thời gian thực thi của bài lập trình. Hệ thống sẽ đo thời gian thực thi cho từng bộ test. Vấn đề này thường hay bị bỏ qua khi làm thủ công. Bên cạnh đánh giá tính chính xác của kết quả thì tiêu chí về thời gian thực thi tương đối quan trọng giúp cho sinh viên cải tiến thuật toán trong bài làm trở nên tối ưu hơn. Đồng thời, nó có thể dùng cho việc xếp hạng sinh viên trong các cuộc thi. Như đã đề cập ở trên, hệ thống hỗ trợ đa ngôn ngữ, tạo điều kiện cho việc giảng dạy nhiều môn học khác nhau. Khi áp dụng vào thực tế nếu cần bổ sung thêm các ngôn ngữ lập trình mới, hệ thống chỉ cần tích hợp thêm compiler tương ứng với ngôn ngữ đó.

5. Kết luận và hướng phát triển

Bài báo đã đề xuất và xây dựng một hệ thống tự động chấm bài lập trình của sinh viên. Qua thời gian triển khai, hệ thống đã mang lại nhiều kết quả khả quan. Đặc biệt, hệ thống đã hỗ trợ rất tốt cho sinh viên trong

quá trình học lập trình cũng như tham gia ôn luyện Olympic Tin học và đạt được thành tích cao trong kỳ thi Olympic Tin học Sinh viên toàn quốc vừa qua. Bên cạnh những lợi ích đã được đề cập ở phần đầu, tác giả nhận thấy tinh thần học tập và thi đua của sinh viên được nâng cao rõ rệt.

Tuy nhiên, để hạn chế số bài điểm 0 do lỗi nhập xuất của sinh viên. Trong tương lai, tác giả sẽ nghiên cứu, bổ sung thêm các tính năng nhắc nhở hoặc phát hiện và thông báo ngay trước khi sinh viên nộp bài□

Tài liệu tham khảo

- [1] Enstrom, E., Kreitz, G., Niemela, F., Soderman, P., Kann, V.: Five Years with Kattis – Using an Automated Assessment System in Teaching. *In Proc. FIE Conference*, IEEE (2011).
- [2] Ihantola, P., Ahoniemi, T., Karavirta, V., & Seppälä, O. (2010, October). Review of recent systems for automatic assessment of programming assignments. *In Proceedings of the 10th Koli Calling International Conference on Computing Education Research* (pp. 86-93). ACM.
- [3] Wang, T., Su, X., Ma, P., Wang, Y., & Wang, K. (2011). *Ability-training-oriented automated assessment in introductory programming course*. *Computers & Education*, 56(1), 220-226.
- [4] Pieterse, V. (2013, April). Automated assessment of programming assignments. *In Proceedings of the 3rd Computer Science Education Research Conference on Computer Science Education Research* (pp. 45-56). Open Universiteit, Heerlen.
- [5] Polito, G. and Temperini, M., 2018. *A gamified approach to automated assessment of programming assignments*. In *Challenges and Solutions in Smart Learning* (pp. 3-12). Springer, Singapore.
- [6] Cerioli, M., & Cinelli, P. (2008). GRASP: Grading and Rating ASsistant Professor. *In Proceedings of the ACM-IFIP IEEEIII 2008 Informatics Education Europe III Conference*. Venice, Italy.
- [7] Xin Chen, Brent Francia, Ming Li, Brian Mckinnon, and Amit Seker. (2004). *Shared information and program plagiarism detection*. *Information Theory, IEEE Transactions on* 50, 7.

Ngày nhận bài: 6/9/2019

Ngày chuyển phản biện: 11/9/2019

Ngày hoàn thành sửa bài: 3/10/2019

Ngày chấp nhận đăng: 10/10/2019