

CHANGES IN LABVIEW PROGRAMS POSTED TO AN ONLINE FORUM AS USERS GAIN EXPERIENCE

Christopher Scaffidi

School of Electrical Engineering and Computer Science, Oregon State University, US

ABSTRACT

Engineers and others can learn to use programming environments such as LabVIEW via online resources, including the LabVIEW forum. However, an interesting challenge in such a diffuse and distributed learning environment is assessing to what extent engineers are increasing in programming skill. This paper presents an analysis exploring the extent to which users' uploaded programs changed in frequency and complexity over time. This study revealed a high rate of drop-out, a drop in the complexity of programs uploaded to the forum during the first two years after users' first (respective) uploads of programs to the forum, and a slow long-term upward trend in complexity. The results highlight the need for further research aimed at assessing and promoting online learning of programming.

KEYWORDS

Web-based Education Systems and Learning Applications

1. INTRODUCTION

Software programming plays an increasing role in engineering practice for instrument automation, modelling and simulation, and other tasks. As a result, many engineers can benefit from learning to program [1]. Although the engineering professions recognizes the need for lifelong learning, and some states even mandate continuing education for some disciplines [2, 3, 4], work-life constraints present a need to integrate ongoing learning with career and other life pursuits [5]. While some practicing engineers have the time and opportunity to enrol in traditional full-time university programs, others instead continue their education part-time and/or off-campus, often through e-learning [6, 7].

These learning opportunities extend to the training of engineers in software programming, for example in how to use the LabVIEW programming environment that is popular in engineering and science. LabVIEW offers a visual language and a massive library of built-in programming functions/primitives for creating programs (called "VIs") that implement data acquisition, data analysis, presentation of information, and other applications [8]. Engineers can learn to program in LabVIEW via online virtual laboratories [9, 10], remote-control laboratories [11], books [12, 13], and an online forum offered by National Instruments (makers of LabVIEW) since 1999 [14]. Programmers can ask one another questions via the forum, post answers, and share and receive code.

An interesting challenge in such a diffuse and distributed learning context is assessing to what extent engineers are increasing in programming skill. As they gain experience with LabVIEW, are they capable of creating and understanding increasingly complex programs? Do users use an increasing proportion of the available language and its associated primitives, or as time passes, do they learn nothing new? In a formal instructional setting (whether in the classroom or online), teachers can evaluate learning by assigning homework, programming problems and exams. An

environment like the online forums, where so many people engage in self-directed learning, offers no opportunity for compelling users to take such tests of mastery.

The example of the Scratch animation environment [15] illuminates a methodological path out of this quandary. As with LabVIEW, Scratch offers an online environment where programmers—who are typically teenagers [16]—can share code with one another. They can also discuss one another's programs, as well as ask and answer questions [17, 18]. Analysing the series of programs that each user has uploaded over time can reveal insights into whether learning may have occurred [19, 20, 21], as well as areas where users have bad habits and may benefit from additional learning [22, 23]. One caveat is that it is important to analyse more than a few hundred users' contributions: one small-scale study focused on users' first 18 months of uploads found that programs decreased in complexity over time [20], whereas a replicated study of many more users' data over a longer period confirmed a slow but definite increase in programs' complexity [19].

The LabVIEW forums thus invite a similar analysis, presented in this paper, aimed at exploring the extent to which users' uploaded programs changed in complexity over time. In addition, the forum data also admit an analysis of whether users typically have stopped participating too quickly for much learning to take place—an extremely common occurrence in the Scratch environment [19, 20]. The following two research questions formalize these objectives:

RQ1: To what extent do VI posts to the LabVIEW forum change in frequency over time, as users increase in experience?

RQ2: To what extent do VIs posted to the LabVIEW forum change in complexity over time, as users increase in experience?

The remainder of this paper is organized as follows. Section 2 summarizes the materials afforded by the LabVIEW forum, as well as the methods used to analyse the data. Sections 3 and 4 present and discuss results, respectively. Finally, Section 5 offers conclusions and opportunities for future work.

2. MATERIALS AND METHODS

A program was written to retrieve all messages posted to the LabVIEW forum since its inception in 1999. For each, the author username was identified (i.e., the username of the account responsible for posting the message), the timestamp, and links to attachments. The program then retrieved each attachment and tested whether it was a VI. If so, it analysed the VI's content to identify what primitives the VI invoked. Any VI that could not be analysed, or which appeared to be empty, was discarded. This yielded a total of 90891 VIs.

Two measures of complexity used in prior research when analysing program complexity [20] were computed for each VI. The first of these, Depth, equalled the total number of primitives comprising the VI. For example, a primitive could include a loop operation, a file-open operation, a socket-write operation, or a call to a sub-VI. The second measure, Breadth, equalled the total number of distinct primitives comprising the VI. Thus, for example, a VI with 1 file-open operation, 13 file-write operations, and 1 file-read operation would have a Depth of 15 and a Breadth of 3.

As in prior work [20], month 0 for a given user was defined as the month in which that person first posted a program to the forum. Month 1 was then the next month, and so forth. Thus, numbering months in this manner temporally aligned VIs according to how much their respective authors had at the time that they posted the VIs to the forum.

Further replicating the method used in analyses of animations, the Depth and Breadth measures were each averaged on a per-month basis, thereby revealing trends in each measure as a function of time (for RQ2). In addition, the counts of VIs and active users were computed for each month (for RQ1), where a user is defined as “active” if he or she posted at least one VI in a given month. Table 1 summarizes measures.

Finally, regressions were performed to answer each of the research questions above. In general, a linear regression of the variable of interest (e.g., number of VIs posted, average Depth, or average Breadth) was performed against time (measured in months since the respective author’s first VI post as explained above). In cases where the regression accounted for less than one-half of the overall variance ($R^2 < 0.5$), additional regressions were performed using exponential and power-law models, to assess whether either of those corresponded better to the actual functional form manifested by the data. Further regressions of a similar nature were performed, as Section 3 discusses in detail, to investigate changes in the frequency and complexity of VIs posted during subintervals of the data set’s time period (e.g., to examine changes in Breadth and Depth during the first 24 months, and/or during the first 150 months, versus during the time period as a whole).

Table 1. Variables used for statistical analyses.

Variable	Definition
Month	Time since a given VI’s author posted a VI to the forum for the first time, numbered from month 0 (primary independent variable).
Number of VIs	Equal to the count of VIs posted in a given Month
Number of Active Users	Equal to the count of users who posted at least one VI in a given Month
Depth	Average VI size for a given Month, measured in number of primitives
Breadth	Average number of distinct primitives in VIs posted during a given Month

3. RESULTS

3.1. User activity as a function of time since each user first posted a VI

The number of VIs posted on a per-month basis dropped sharply from a total of 23934 in users’ first month of posting VIs (month 0) to only 6343 in the next month (month 1), with the drop-off soon flattening through month 150 (Figure 1). Thereafter, the number of VIs posted again fell sharply. Consequently, even though the midpoint of the period was 75 months, in practice the average value for Months across all VIs was only 28.1 months.

Most of the drop was attributable to the decrease in the number of active users. In particular, the fall in the number of users posting VIs followed a sharp drop-off comparable to that of VIs (Figure 2), with a steep fall during the first few months, a flattening thereafter, and a sharp collapse after month 150.

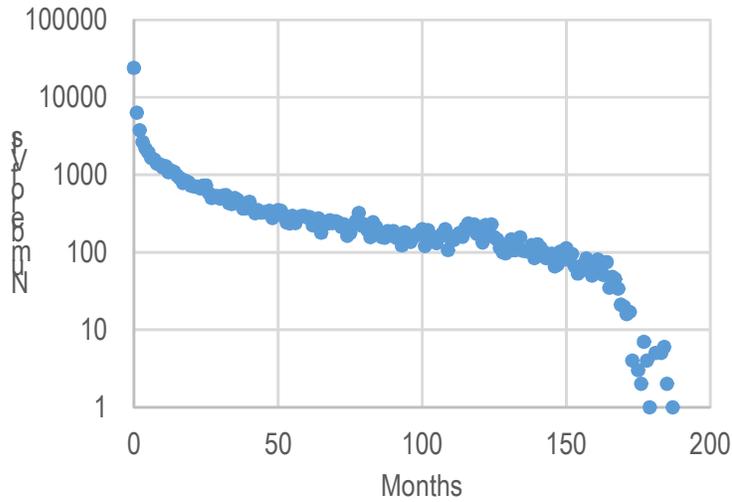


Figure 1. Distribution of VIs in the LabVIEW forum, where month 0 is the first month in which each VI's author first posted a VI to the forum.

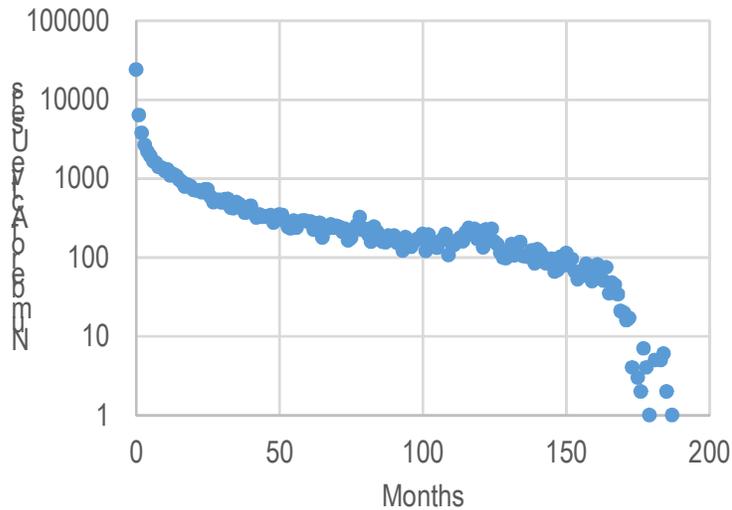


Figure 2. Number of users who posted VIs in each month after each user's first posted VI.

A power law closely described the number of active users through month 150 (Eq. 1), with $R^2 = 0.98$. The regression indicated that the number of active users was approximately inversely proportional to the number of months that had passed since each user's first posted VI.

$$\text{Number of active users (in each of the first 150 months)} \approx 4928 \cdot (\text{month} + 1)^{-0.97} \quad (\text{Eq. 1})$$

Based on this fit, the number of users still active after 2 years (24 months) was approximately 4% of the initial active user population, that after 4 years was approximately 2% of the initial population, and that after 8 years approximately 1%.

It is worth noting that the time-range shown in Figure 2 comprises the lifetime of the LabVIEW forum. It is conceivable that relatively few people have been active for a long time simply because, although they are still active, they only recently joined the forum. One way to test this explanation for the apparently high drop-out rate is to perform a cohort analysis, examining only

uploads of users who joined the forum in a certain year. Selecting 2009 is reasonable, for example, as a major LabVIEW release occurred that year. Nonetheless, replicating the analysis above using only data from the 486 people who first posted VIs in 2009 revealed a similar drop-off in activity—in this case, the number of active users fell off even faster than inversely proportional to the number of months since each user’s first posted VI (Eq. 2).

In contrast, the number of posts per user did not display any sharp drop-off until after month 150 (Figure 3). If anything, the number of VIs posted per person increased slightly through month 150, as linear regression yielded a slight upward slope (Eq. 3), although this fit accounted for little between-month variance ($R^2 = 0.05$). Thus, in summary, the overall drop in VIs posted to the forum (Figure 1) was largely attributable to the drop in active users (Figure 2) rather than due to a reduction in per-user activity (Figure 3).

$$\text{Num. active users (among those who first posted VIs in 2009)} \approx 1902 \cdot (\text{month} + 1)^{-1.17} \quad (\text{Eq. 2})$$

$$\text{Num. posted VIs per user (in each of the first 150 months)} \approx 0.0019 \cdot \text{month} + 2.74 \quad (\text{Eq. 3})$$

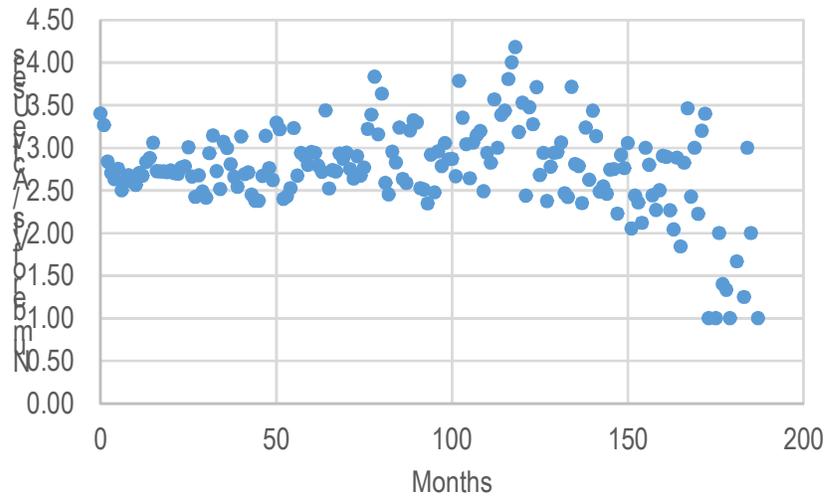


Figure 3. Number of VIs posted per user, versus months since each user’s first posted VI

3.2. Depth and Breadth demonstrated by users’ posted VIs

The Depth and Breadth measures were each averaged over VIs posted in each month since the corresponding author’s first posted VI. Depth generally remained between 20 and 40 primitives per VI (Figure 4), while Breadth was typically between 11 and 15 distinct primitives per VI (Figure 5). Both measures demonstrated increased scatter after month 150, as is expected due to the sharp decline in the number of VIs for those months (as discussed in Section 3.1). In addition, Depth exhibited an order of magnitude more scatter throughout the period than Breadth ($\sigma_{\text{Depth}}=19.20$ versus $\sigma_{\text{Breadth}}=1.79$).

Both measures also demonstrated slight upward trends over the duration of the data set. Linear regression revealed that Depth rose at an average of 0.02 primitives per month, while Breadth rose more slowly at 0.01 distinct primitives per month (Eqs. 4a, 4b). This very slight increase over time only accounted for a small fraction of the overall variance ($R^2=0.007$ for Depth, $R^2=0.21$ for Breadth), indicating that this change over time was relatively small to other month-to-month changes.

Average Depth (in each of the first 150 months) $\approx 0.02 \cdot \text{month} + 28.27$ (Eqs. 4a, 4b)

Average Breadth (in each of the first 150 months) $\approx 0.01 \cdot \text{month} + 11.75$

Interestingly, the scatterplots (Figure 4, Figure 5) both demonstrate an interesting drop in the corresponding measure during the first two years of use. Depth fell from 33.7 in month 0 to only 22.7 in month 23: that is, VIs posted in users' first month were nearly 50% larger than VIs posted two years later. Breadth fell a smaller but non-negligible amount from 13.3 to 11.1 in the same time range. Moreover, a linear fit for each measure over the 2-year period (Eqs. 5a, 5b) indicated that this trend accounted for the majority of the scatter in month-to-month changes of these two measures ($R^2=0.74$ for Depth, $R^2=0.66$ for Breadth).

Average Depth (in each of the first 24 months) $\approx -0.42 \cdot \text{month} + 33.69$ (Eqs. 5a, 5b)

Average Breadth (in each of the first 24 months) $\approx -0.08 \cdot \text{month} + 13.25$

These two downward-trending slopes were an order of magnitude larger than the corresponding two upward-trending slopes for the entire multi-year period (Eqs. 4a, 4b): both measures fell more in the first two years than they subsequently rose in the entire remainder of the dataset.

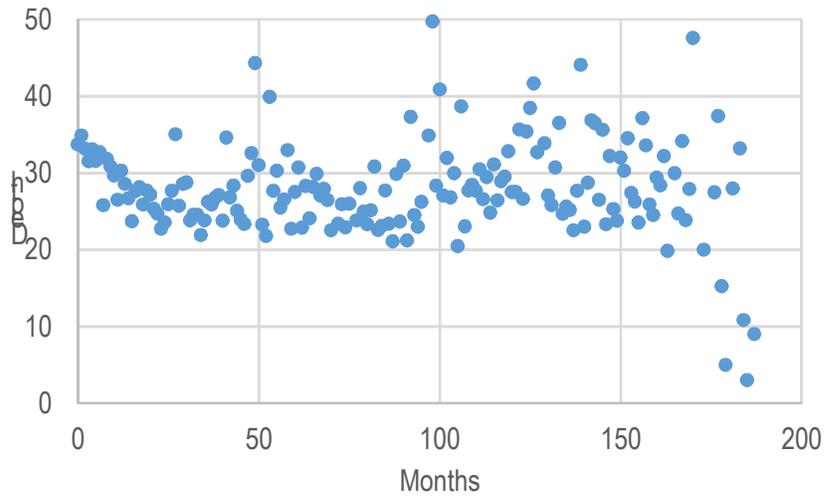


Figure 4. Average VI Depth, as a function of time since each users' first posted VI

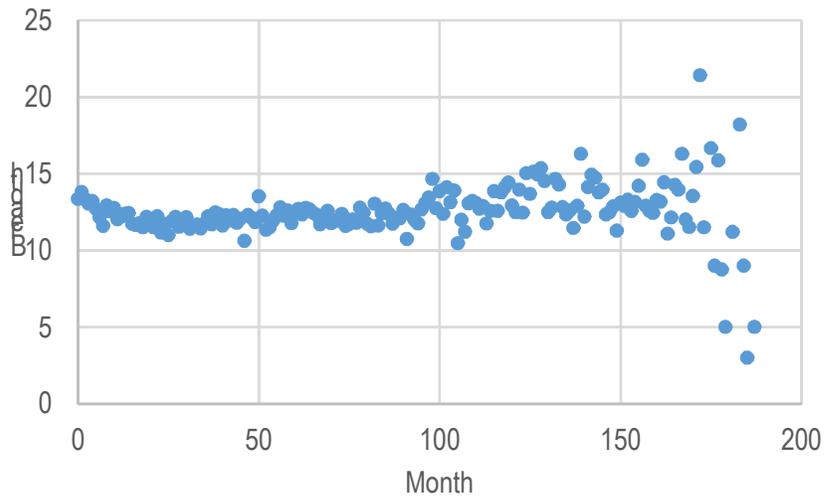


Figure 5. Average VI Breadth, as a function of time since each users' first posted VI

3.3. Replicated analysis of Depth and Breadth in first two years using only data from longer-term users

Given the steep dropout of users during the first two years (Section 3.1), perhaps the most straightforward explanation for the differing long-term and short-term trends in Depth and Breadth (Section 3.2) could be that user abandonment of LabVIEW skewed data for the initial 2-year interval. Note that for this to account for the fall in Breadth and Depth over those 2 years, it would be necessary for users at month 0 to be posting more complex VIs than those still active at month 23: in other words, the users contributing relatively complex VIs would need to drop out faster than users contributing relatively simple VIs.

All VIs were filtered to include only those from users who were still active on or after month 25. Thus, VIs were discarded from any user who left the forum during months 0 through 24. Then, average Depth and Breadth were then recomputed for months 0 through 23, and the linear fit was then repeated for each measure (Eqs. 6a, 6b). The resulting slopes were again negative, albeit not so large in magnitude as the corresponding slope among all users for the first 2 years (Eqs. 5a, 5b), yet nonetheless quite a bit larger in magnitude than the longer-term slopes (Eqs. 4a, 4b).

Average Depth (non-dropouts, each of first 24 months) $\approx -0.26 \bullet \text{month} + 29.10$ (Eqs. 6a, 6b)

Average Breadth (non-dropouts, each of first 24 months) $\approx -0.05 \bullet \text{month} + 12.36$

The implication is that users who persisted in posting VIs beyond two years may, indeed, have exhibited somewhat different progression than those who left the forums prior to the two-year mark, although drop-out by itself does not fully account for the sharp drop in both Depth and Breadth during the initial two-year interval.

4. DISCUSSION

The results address the research questions that motivated this work:

RQ1: The number of VIs decreased sharply, primarily due to a corresponding decrease in the number of active users. The likelihood of a given user being active was approximately inversely proportional to the number of months since the user had first posted a VI to the LabVIEW forum, with only 2% (approximately) still active after two years.

RQ2: The complexity of VIs very slowly increased over the time period of the data set in terms of both Breadth and Depth. However, this long-term rise followed a sharp drop in both measures during the first two years after users posted their first VIs. This initial decrease remained evident, though less sizable, even when restricting the analysis only to users who remained active beyond the 2-year period, suggesting that user attrition cannot fully explain the drop in these measures of complexity.

These results share some similarities to those of the prior studies of animation programs posted to an online repository. As in prior research focused on animation programmers [19, 20], the LabVIEW forum demonstrated an extremely high drop-out rate, with well over half of users no longer remaining active after 2-3 months. Moreover, the initial study of animation programs also found that program complexity decreased [20]; in that study, only 4% of the programs shared by 250 distinct users were posted more than 18 months after the respective authors' first post, so the observed downtrend had little empirical support from programmers having more than 2 years of experience. In contrast, the replicated study with more than two orders of magnitude more users was able to more extensively incorporate information about programs posted by more experienced users [19], and this study found a very slight upward trend. Likewise, the current

study indicates a sharp decrease in VI complexity posted by relatively less experienced users, superimposed on a less steep increase in complexity over a longer term. Another point of comparison is a third study of the Scratch animation repository that clustered users according to characteristics of their learning trajectories, which revealed that only 357 of the 3852 people examined actually fell into the cluster of users who demonstrated the highest rate of learning [24]. The current study analogously raises concerns about the rate at which LabVIEW users attain high rates of learning by means of the online support environment. Given the relatively flat slope of increase in program complexity, it is possible that LabVIEW users are learning over time, but the available evidence at this time indicates that any such learning is slow.

Unlike the first animation-program study, but rather like the later work, the current study employed a whole-population analysis. Thus, it is possible to rule out “unlucky” sampling as the cause of these phenomena in the LabVIEW environment. Future work should instead focus on investigating other potential explanations of the observed phenomena, as outlined in Section 5, as well as addressing the high rate of dropout and apparent absence of a high rate of learning among users.

5. CONCLUSION

Overall, this study has revealed that users in the LabVIEW forum stand a high probability of becoming inactive within one year of initially posting a VI, that the VIs they post to the forum decrease in complexity over the initial two years, and that the VIs posted then slowly become increasingly complex. Future work could build on this study in several areas.

First, additional research could investigate potential causes of the curious drop in VI complexity during the first two years. For example, perhaps users initially post very large VIs that they themselves find on the web but do not understand, and as they gain experience, they become more discerning about what VIs (or what portions of VIs) to post on the forum. Or perhaps they rapidly become more efficient in their use of primitives, such that the code that they create for themselves (and then post to the forum) uses fewer primitives and less diverse primitives. Dropout effects may account for some of the observations, as well. Future studies of the forum, perhaps coupled with interviews of users, could aim to reveal the extent to which these and other factors explain different aspects of the current study’s findings.

Second, this study invites work aimed at stemming the massive drop-out rate among users. Additional research could investigate the extent to which users who leave the forum continue to use LabVIEW, whether and how they continue learning about programming and about LabVIEW in particular, what challenges slow or block their learning, and how to help people overcome those barriers.

Finally, future work could develop additional measures of VI complexity that inform a deeper understanding of user proficiency. Already, research has begun developing trajectory-based measures of the programming primitives and supporting concepts that animation programmers have mastered [24]. Adapting and validating these approaches for use in LabVIEW and other programming environments could facilitate cohort-based analyses of learning. In particular, such measures could play a starring role in evaluations of whether new versions of the LabVIEW environment are more effective at supporting learning than older versions, thereby helping to guide the development and refinement of the tool suite and online support systems.

ACKNOWLEDGEMENTS

National Instruments funded this research and gave permission to download the contents of the online forums. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of National Instruments.

REFERENCES

- [1] Guo, P. (2013) Why scientists and engineers must learn programming. Communications of the ACM Online Blog, <http://cacm.acm.org/blogs/blog-cacm/166115-why-scientists-and-engineers-must-learn-programming/fulltext>
- [2] Guest, G. (2006) Lifelong learning for engineers: A global perspective. *European Journal of Engineering Education*. 31(3), 273-281.
- [3] Lenschow, R. (1998) From teaching to learning: A paradigm shift in engineering education and lifelong learning. *European Journal of Engineering Education*. 23(2), 155-161.
- [4] Martinez-Mediano, C., and Lord, S. (2012) Lifelong learning competencies program for engineers. *International Journal of Engineering Education*. 28(1), 130-143.
- [5] Gonzalez, C. (2016) The value of a continuing engineering education. *Machine Design*, <http://machinedesign.com/learning-resources/value-continuing-engineering-education>
- [6] Bourne, J., Harris, D., and Mayadas, F. (2005) Online engineering education: Learning anywhere, anytime. *Journal of Engineering Education*. 94(1), 131-146
- [7] Ubell, R. (2000) Engineers turn to e-learning. *IEEE Spectrum*. 37(10), 59-63.
- [8] Jamal, R., and Wenzel, L. (1995) The applicability of the visual programming language labVIEW to large real-world applications. *IEEE International Symposium on Visual Languages*, 99-106.
- [9] Basher, H., and Isa, S. (2006) On-campus and online virtual laboratory experiments with LabVIEW. *IEEE SoutheastCon*, 325-330.
- [10] Ertugrul, N. (2000) Towards virtual laboratories: A survey of LabVIEW-based teaching/learning tools and future trends. *International Journal of Engineering Education*. 16(3), 171-180.
- [11] Orduna, P., Garcia-Zubia, J., Rodriguez-Gil, L., Irurzun, J., Lopez-de-Ipina, D., and Gazzola, F. (2012) Using LabVIEW remote panel in remote laboratories: Advantages and disadvantages. *IEEE Global Engineering Education Conference*, 1-7
- [12] Bishop, R. (2014) *Learning with Labview*, Prentice Hall.
- [13] Travis, J., and Kring, J. (2006) *LabVIEW for Everyone: Graphical Programming Made Easy and Fun*, Prentice Hall.
- [14] LabVIEW forums, National Instruments, <http://forums.ni.com/t5/LabVIEW/bd-p/170>
- [15] Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., and Rosenbaum, E. (2009) Scratch: Programming for all. *Communications of the ACM*. 52(11), 60-67
- [16] Lee, Y. (2011) Scratch: Multimedia programming environment for young gifted learners. *Gifted Child Today*. 34(2), 26-31.

- [17] Dahotre, A., Zhang, Y., and Scaffidi, C. (2010) A qualitative study of animation programming in the wild. ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, 29-38.
- [18] Scaffidi, C., Dahotre, A., and Zhang, Y. (2012) How well do online forums facilitate discussion and collaboration among novice animation programmers? ACM Technical Symposium on Computer Science Education, 191-196.
- [19] Matias, J., Dasgupta, S., and Hill, B. (2016) Skill progression in scratch revisited. ACM Conference on Human Factors in Computing Systems, 1486-1490.
- [20] Scaffidi, C., and Chambers, C. (2012) Skill progression demonstrated by users in the Scratch animation environment. International Journal of Human-Computer Interaction. 28(6), 383-398.
- [21] Xie, B., and Abelson, H. (2016) Skill progression in MIT App Inventor. IEEE Symposium on Visual Languages and Human-Centric Computing, 213-217.
- [22] Boe, B., Hill, C., Len, M., Dreschler, G., Conrad, P., and Franklin, D. (2013) Hairball: Lint-inspired static analysis of Scratch projects. ACM Technical Symposium on Computer Science Education, 215-220.
- [23] Moreno, J., and Robles, G. (2014) Automatic detection of bad programming habits in Scratch: A preliminary study. IEEE Frontiers in Education Conference, 1-4.
- [24] Yang, S., Domeniconi, C., Revelle, M., Sweeney, M., Gelman, B., Beckley, C., and Johri, A. (2015) Uncovering trajectories of informal learning in large online communities of creators. ACM Conference on Learning@ Scale, 131-140.

AUTHOR

Christopher Scaffidi earned a Ph.D. in software engineering from Carnegie Mellon University and is currently an associate professor of computer science in the School of EECS at Oregon State University. His research interests are where human-computer interaction and software engineering intersect. Most of his current projects aim to help software users to create code for themselves, and to effectively share that code with one another.

