Exploring the Impact of Video on Inferred Difficulty Awareness

Jason Carter[1], Mauro Carlos Pichiliani[2], Prasun Dewan[3]

[1] Cisco Systems-RTP
North Carolina, U.S.A.

[2] IBM Research AI
São Paulo, Brasil

[3]Department of Computer Science – University of North Carolina-Chapel Hill
North Carolina, U.S.A.

jasoncartercs@gmail.com, mpichi@br.ibm.com, dewan@cs.unc.edu

Abstract. *This paper explores difficulty awareness, which is motivated by academic and industrial collaboration scenarios in which unsolicited help is offered to programmers in difficulty. We performed experiments to determine how well difficulty can be automatically inferred by mining the interaction log and/or videos of programmers. Our observations show that video combined with other logs can be employed to infer difficulty. These results imply that (a) when collaborators can be seen, either directly or through a video, posture changes, though idiosyncratic, are important cues for inferring difficulty; (b) automatically inferred difficulty, using both interaction-logs and postures, when possible and available, is an even more reliable indication of difficulty; (c) video can play an important role in providing unsolicited help in both face-to-face and distributed collaboration.*

1. General Information

Awareness technology, like technology supporting direct collaboration, has supported sharing of collaborator state captured by multiple media. Moreover, awareness technology, like technology supporting direct collaboration, can attempt to give users the feeling of "being there" in one location or go "beyond being there" by supporting forms of sharing not directly provided by face to face interaction.

One form of awareness supported is whether a remote user is facing difficulty. An important reason for awareness is to determine if collaborators are in difficulty and if help is necessary. Making the computer infer difficulty can reduce the amount of information to be transmitted to a remote helper and/or relieve the helper from manually determining if collaborators are in difficulty, thereby allowing the helper to discover and process difficulties of a larger number of users.

Our initial motivation for this work comes from previous research that shows that as distance among programmers increases, there are fewer opportunities to offer help to team members and the productivity of programmers decreases [Herbsleb, Mockus et al. 2000; Teasley, Covi et al. 2000]. We believe that inferred difficulty

awareness can be used instead of physical or computer-supported awareness of collaborator's activities in both industrial and educational environments. Moreover, the potential helpers can be working exclusively on the task of providing help or interrupt their task to provide help based on inferred difficulty from cues. In our context difficulty is based on programming barriers that block, hinders or somehow affect how developers solve their tasks. For instance, the lack of knowledge about the parameters of a function is a barrier that we address in this research.

We have iteratively developed an Eclipse extension that automatically detects programming difficulty, communicates this information to interested observers, and allows the observers to offer help taking into consideration the difficulty barrier type, which include the surmountable and unsurmountable programming task gathered from user field study. In the study distributed students were offered help with their homework in response to automatically detected difficulties.

Given that, by definition, difficulty is a rare event (otherwise the worker and activity are mismatched), it makes it easier for helpers to find difficulty in collaborator states, thereby making it easier for them to support a larger number of remote collaborators in difficulty. Moreover, this work has explored how a state-of-the-art difficulty-inference algorithm should be changed if rare difficulty events searched include not only the interaction logs but also the videos of the workers. We choose to focus only in programming logs and video posture. However, other aspects, such as automatic emotion, personalities, programming style and context, could also be employed to improve the detection of difficulty barriers.

Our work has drawn from previous research in multimodal/multimedia emotion detection [D'Mello and Kory 2012]. Our results strengthen previous findings [Carter and Dewan 2018] that show that (a) multimodal effect detection is superior to unimodal detection, and (b) the predictive power of postures depends on the personality of the subject.

2. References

D'Mello, S. and Kory, J. (2012) "Consistent but modest: a meta-analysis on unimodal and multimodal affect detection accuracies from 30 studies", In: Proceedings of the 14th ACM international conference on Multimodal interaction (ICMI), ACM, New York, NY, USA.

Herbsleb, J. D., Mockus, A., Finholt, T. A. and Grinter, R. E. (2000) "Distance, dependencies, and delay in a global collaboration", In: Proceeding of the 2000 ACM conference on Computer supported cooperative work, Philadelphia, Pennsylvania, USA, pp. 319-328.

Teasley, S., Covi, L., Krishnan, M. S. and Olson, J. S. (2000) "How does radical collocation help a team succeed?", In: Proceedings of the 2000 ACM conference on Computer supported cooperative work, Philadelphia, Pennsylvania, USA, pp. 339-346.

Carter, J. and Dewan, P. (2018) "Contextualizing inferred programming difficulties", In: Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering, Gothenburg, Sweden, pp. 32-38.