We appreciate very much the reviewers’ constructive reviews!

===

Reviewer-1 wonders about utility of TeMAPI in practice, since current translation tools include mapping for only a small subset of J2SE and .NET APIs and our results show that 70-80% of code cannot be translated. There are three major aspects listed below that need to be taken into consideration and clarified here.

First, we apologize for giving a wrong impression regarding the utility of translation tools with the result of 70-80% code cannot be translated. These numbers are related to wrapper code, which is generated for each API method of J2SE and .NET libraries, and do not mean or imply that 70-80% of application code in practice cannot be translated (Reviewer-1 seems to have such wrong impression).

Second, although these tools support mapping for only some APIs, these supported APIs are the most commonly used, and these tools’ significant utility/popularity is reflected by their high download counts. Any TeMAPI-detected differences among those supported APIs can cause significant hard-to-detect defects. Orthogonally, [30] exists to provide mappings for more APIs. Furthermore, as shown by our previous work “Thummalapenta and Xie. SpotWeb: Detecting Framework Hotspots and Coldspots via Mining Open Source Code on the Web. ASE 2008”, only a few APIs (5%~19%) provided by libraries are often used by programmers (referred to as hotspots). We identified that the APIs supported by translation tools often fall into this hotspot category.

Third, all reviewers are convinced with the importance of the problem of language migration in practice, as described in the first two paragraphs of the introduction of our paper. To address this important problem, besides being applicable on those API mappings (from existing translation tools) that induce compliable code, TeMAPI is still applicable in scenarios where translation tools could not translate successfully and programmers manually fix syntactically-wrong API mappings to get rid of those compilation errors; there, TeMAPI could be applied against manually-fixed API mappings, helping programmers increase their confidence on the correctness of the translated code.

===

Reviewer-1 says that the paper is repetitive and includes grammatical issues. We assure that we will address all those issues in our final version.

===

Reviewer-3 concerns about the difference between TeMAPI and previous work (i.e., MAM [30] and Doc2Spec [31]). MAM [30] and Doc2Spec [31] are both orthogonal to TeMAPI in their research questions and approaches. Still, these tools can benefit each other. In particular, the results revealed by TeMAPI can help MAM mine better API mapping relations that can fix more behavioral differences, and we can integrate Doc2Spec with TeMAPI to detect more behavioral differences that are related to call sequences. We will add more discussions in our final version.

===

Reviewer-3 says that TeMAPI is related to Quest, since both use a random approach for generating tests. However, besides random testing, TeMAPI also uses state-of-the-art test generation technique called dynamic symbolic execution. We will add discussion of Quest to related work and describe more details on how Quest is different from TeMAPI.

===

Reviewer-3 wonders about the availability of the tool. We will release our tool soon through our project website.