Paper review of the entitled research paper « iPOJO : an Extensible Service-Oriented Component Framework »

Anis HARRAK

24 mars 2015

1 SUMMARY :

Firstly, after reading carefully this research paper, I think that the research problem the paper attempts to address is the fact that there are some weaknesses in the existing approaches allowing dynamism in applications. The main weakness here is that existing services-oriented computing models focus only on services interaction and dont care about other non-functional properties management such us persistency, security, configuration, management, eventing...Thus, the paper is trying to solve the insufficiency of these existing paradigms and then to deal with complex and dynamic requirements of applications.

As a solution to this issue, the paper proposes a new approach to simplify the development of dynamic service applications ; the service-oriented component iPOJO which is a component model managing dynamic service interactions and providing an extensibility mechanism.

Accordingly, to substantiate their claims, authors mentioned some advantages of the new proposed approach :

- It will accelerate decision-making software with specific, formatted and up-to-date informations.
- It will enable faster reaction with changes.
- It will promote the establishment of innovative E-services.

Furthermore, before presenting the new framework iPOJO, service-oriented component models looked only on services interaction. Whereas, in introducing the extensible service-oriented component IPOJO, model could (in addition to the fact of focusing on services interaction) also extend the standard features with non-functional property management like persistency, security

Finally, with its power, simplicity and non-intrusiveness, iPOJO became the best solution to achieve both extensibility and dynamism in applications.

2 EVALUTION :

Nowadays, several component models exist ; they aim different application domains. They repose mainly on the developer effort to integrate their business logic into the component model. Furthermore, these models dont support service dynamism. Consequently, the development of dynamic application remains a hard task for developers and needs more and more efforts to fulfill complex and dynamism requirements. Therefore, the authors of this research paper are solving these problems and exploring a new approach which deals with both services interaction issue and extending standard features with other non-functional property management such as persistency, security, eventing

Thence, I think therefore that the paper is worth reading because it targets a well-known open problem by exhibiting its insufficiencies and complexities. Thus, after looking well at the existing approaches features, authors highlight the new approach as a solution to the functionalities limitation of existing models.

Ultimately, I think that claims proposed by the authors are valid and proved without problematic experimental setup. We behold this when we found that proposed approach was evaluated in an industrial prototype for home control within the ANSO ITEA European research project. Furthermore, it was also demonstrated at IEEE CCNC and hosted as a subproject of the Apache Felix project.

3 SYNTHESIS :

Throughout the entire paper, the authors tried to solve the problem of dynamic component availability. In fact, the existing service-oriented component models target only the communication between services, which is insufficient to support dynamism and extensibility. Indeed, they should support other non-functional property management like persistence and security

As a solution to address the research issue, the authors proposed the service-oriented component model. Thus, they introduced the iPOJO framework including its flexibility mechanism. Then they discussed some implementation details of the framework and finally, they illustrated a usage of iPOJO in a specific domain.

In my point of view, a better way to substantiate the authors' claim otherwise, is to start with describing the framework as an alternative approach. Then, linking up with the most important framework weaknesses .And finally, taking these defaults one by one and tries to build qualities and advantages from. I think it will be better idea than focusing only on advantages. As I said previously, the best way to substantiate claims is to work more on weaknesses in purpose to strengthen our ideas.

However, I think that the solution proposed by authors doesnt resolve the problem completely. By changing applications dynamically, we can encounter probably, some inconsistencies leading to failure. So, to improve the research results, we should manage the runtime monitoring to ensure the dynamic applications consistency.

Although iPOJO resolved the main issue, it remains insufficient until it manages consistency and efficiency despite dynamic changes. Thus, I think we can do better than the author of this paper if we manage the runtime monitoring requirement.

Finally, from my point of view, I think this research will be very helpful for us in our future thesis. Besides, this paper taught us which framework using to solve the problem of dynamism, so, the remaining task is to find and diagnose errors caused by dynamic changes, which is the purpose of our thesis.