nutella - Contributor’s feedback

Matteo Palvarini

1) Imagine you met a developer friend and she was telling you about some new technologies she's recently worked with. She then asks you: "What have you been working on recently?" How would you describe what nutella is and does for you to her?

nutella is a framework that helps developers create applications composed of both a back-end and a front-end and powered by a useful publish-subscribe messaging protocol based on MQTT.

More specifically, in the context of nutella the back-end applications that a developer can create are called ‘bots’ and the front-end ones are called ‘interfaces’. The core idea of nutella is to make the process of creating these applications as easy as possible for the developer: he just has to download the framework and use a few basic commands from command line to create a new project and start it. The developer has just to implement the interfaces and bots, leveraging the power of the communication protocol embedded into nutella and easily accessible through the APIs.

When a new project is created, the framework takes care of automatically generating all the needed hierarchy of folders and files, of starting the web server which serves these files and even to monitor the back-end bots during execution.

Other than the above general capabilities, the framework has been designed to be able to support typical classroom applications, such as the ones created in the context of the learning technologies research. The nutella features in this sense are represented by introducing the distinction between app_id and run_id of different applications and runs of those applications. This solves the frequent need to have the same application run at the same time in different classrooms and with different data. nutella takes care of all this complexity in a way that is almost transparent to the developer and very easy to implement: an application can be simply started multiple times with different run ids and app ids.

A further domain specific capability of nutella is given by the framework-level components that are shipped together with the framework itself. Among these, for example, RoomPlaces, which keeps track of the position of resources in the physical space of the classroom, RoomCast, which acts as provider of a selection of available interfaces for the devices in the classroom and RoomMonitor, which provides developers with a GUI to monitor applications in real-time during both development and enactment.

As the developer of RoomCast, I can say that nutella has been essential to me especially to support the inter-communication between the different components of my system: RoomCast has a very heterogeneous nature, since it is composed by a bot in ruby, a few web interfaces written in javascript and a few iOS applications which use both the javascript and Swift implementations of the nutella lib for communicating. In this sense nutella made my job much easier by providing messaging APIs which effectively work among different languages and keeping a good consistency between these different API calls so that switching from one language to another one is quite immediate.

The second essential feature of nutella which has greatly improved my work is the support for many different runs of the same application: RoomCast is shipped with nutella, but every single nutella app created by a third developer will populate Roomcast with its own data and use it for its own purpose.

Finally, RoomCast also leverages an additional feature of nutella which takes care of managing the upload of images to the server: use cases that include the need to do this are quite common and nutella is able to make even this aspect very quick and intuitive for the developer.

2) Suppose then she asked you: "Cool, but what are the strength and weaknesses of nutella?" How would you answer that question?

As mentioned before, the strength of nutella is its ability in making a developer's life much easier by supporting him throughout the whole development process.
This is achieved through the automatic processes of installing and deploying a new project, through the nutella_lib library, i.e. the one which contains the publish-subscribe messaging protocol, and through the wide number of framework level functionalities shipped with nutella.

Moreover, the nutella_lib library comes in a variety of different languages (and can easily be extended to new languages). This allows for an easy and effective communication between all the components of a nutella application. The most common example is the communication between bots (written in ruby) and interfaces (written in javascript). The main purpose of the library, which wraps the MQTT protocol, is to support inter and intra communication between components of an application to have them update in almost real-time as a consequence of messages sent and received over the network.

Another key aspect of nutella is the modularity with which it has been designed: this allows the framework to be extended at any time. I experienced this when I integrated RoomCast into the framework: a few changes to the RoomCast bot and interfaces allowed it to be shipped with nutella and leverage all the capabilities of a framework level component without having to modify any other existing nutella code.

nutella is continuously growing and improving: many of the original weaknesses have already been fixed based on the feedback of the contributors and the particular needs that they happened to have over time. An example of this is the database storage: if nutella's classical way to do it, through a json file, is sufficient to support RoomCast, for other applications this method turned out to have some limitations so more advanced features such as MongoDB support were added.
I have to say that all these are possible improvements added over time and they don't show structural weaknesses of the framework itself, which instead appears to be solid.
Another possible future improvement could be a more user-friendly interface to guide through the nutella setup without having to use the command line.

Finally, even if the framework is specifically meant to be used in the context of the classroom and for learning technologies applications, I can foresee a much wider range of projects which could leverage the power and simplicity of nutella in many other domains.