

Practical Session 3–Autonomic Computing & Emergent Software Systems

Interactive Assembly

In this practical session, we are going to have our first contact with one of the most important modules that supports the concept of Emergent Software Systems: The **Assembly** module. You can find all code for this practical in our **GitHub** repo in: https://github.com/barryfp/summer_school.

The Assembly module implements the entire assembly process we described in *Lecture 2*. Given a root component, the Assembly module can extract provided and required interfaces from the root component object file, search for components and apply this same process recursively until all components are found, loaded and connected into a fully functioning system. Furthermore, the Assembly module enables the user to query the current running architectural composition and get the list of every possible composition for the software. There is a dedicated lecture to detail the Assembly module functionalities and how they work. For this practical, however, we only want you to familiarise yourself with the Assembly module by using it through a component named **Interactive Assembly**.

The *InteractiveAssembly.dn* is a component that encapsulates the Assembly functions and provide an interactive terminal for the user. You can find all necessary code to use the Assembly module through the Interactive Assembly component in the “*Practical3*” folder inside the “*pal*” subfolder. As we did in previous practical sessions, compile all components in the “*pal*” subfolder and in the “*calculator*” subfolder by running “*dnc .*” in each directory. We are going to run the Interactive Assembly to compose and adapt a calculator program. To execute this example, after compiling all components, go to the “*pal*” folder and type: “*dana -sp ../calculator InteractiveAssembly ../calculator/Calculator.o*”.

The execution of the Interactive Assembly triggers the Assembly module to search for components, load them and wire them. As a result, you will see on the screen a calculator application. Make sure you play with the calculator to make sure it is working (:). After that, you can get a list of the Interactive Assembly functions by typing “*help*” into the command line, as a result you will get a list of commands that the Interactive Assembly can execute for you. Suggestion: try getting a list of possible compositions and change from one composition to another.

Assignment

For this practical assignment, we expect you to add more buttons to the calculator and implement the functionalities for these buttons. You could add another operation (e.g. division, multiplication, etc.), or a button that can clean the display, so we can perform multiple operations without having to restart the calculator program. If this first part of the assignment is too easy for you, you may also try to add component variants for the Button interface, with different colours, sizes so that you end up with more composition choices to choose from in the Interactive Assembly.