DS2 Front End

By Heath J. French

Introduction

The DS2 Front End is the interface used by people to convert Distributed Systems using Akka Actors into Distributed Systems that can be check using the DS2 software. The Front End works by allowing people who have constructed builds of Akka systems using the sbt platform to pass their sbt project as an argument for the Front End. From there, the Front End builds the project, looks for scala files, and then finds all implementations of Actor classes in your project's files. From here, each Akka Actor class is converted into an instance of an Agent class which is used by DS2 to perform its analysis.

Front End Limitations

The Front End has some limitations and this is where we will discuss those. Anything that isn't explicitly mentioned here should work properly otherwise.

SBT library and Runnable

For a program to be parsed using the Front End, it must be in an SBT project and said project should be able to be compiled without any exceptions.

Global Variables

While global variables work with the Front End, global variables cannot be access by using the "*this.global"* syntax. This means that, when local variables exist with the same name, the global variable will be inaccessible at that part of the program.

Recursion

Functions that call themselves currently aren't handled by the Front End.

Calling External Classes

While the Front End can handle instances where external classes are used in an Actor, DS2 can't guarantee the reliability of its results if external classes are used since they are outside of the scope of the DS2 analyzer.

Actor's Receive Match Cases

The Front End currently can only parse different incoming messages if the overridden receive function uses a match syntax to parse incoming messages.

DS2 Front End

By Heath J. French

Use Case Classes for incoming Messages

Messages sent between Actors must be in the form of Scala Case Classes. As long as the Case Classes are publicly accessible, the Front End will be able to see them and then parse them accordingly.

No Multi-Threading in Actors

Both DS2 and the Front End assume that there is no form of parallelism in Actors. If this isn't the case, DS2 will be unable to properly analyze the Distributed System created by the Front End.

Actor Related Functions

Actors can perform the following functions inside of a class:

- become and unbecome
- tell and !
- ask and ?

Any other functions implemented by Actors currently aren't implemented in this version.

Tutorial

The Front End is accessed from a terminal and is run using java. The following arguments can be included when running the Front End.

-parse *[address to sbt project]*

If you want to use the Front End to parse a new Akka System, then including the argument will make sure that the Front End knows where to look. Unless you have previously saved a parsed version of your system, this is the only way to run your Akka System through DS2.

-save *[address where to save parsed system]*

Once a distributed system has been parsed, the Front End will attempt to save the parsed Actors in the specified folder.

DS2 Front End

By Heath J. French

-load *[address where to load parsed system]*

If you already have a parsed system, this options can be used to load your previously parsed system. When this is used, the *–parse* options won't need to be used.

-verbose

Prints out detailed information about the parsing process while it is taking place.

-ignoreExceptions

By default, the Front End will end early if a Distributed System cannot be parsed. Add this argument though, will allow the system to continue to attempt parsing the system. This can be useful if you want a comprehensive idea of all the things that need to be modified in the System for the Front End to be able to parse it.

-quit

Normally, once the Front End is finished parsing a Distributed System, it will allow a user to then start using DS2 to perform analysis on their system. By using this argument though, the Front End will return as soon as parsing (and possibly saving) has taken place.