Integrating CertainTrust into OpenCCE

Nikolaos Alexpoulos, CROSSING S1

August 2016

1 About CertainTrust

CertainTrust is a robust Bayesian Computational Trust model for user and agents. It offers a mathematically sound implementation of a Bayesian probability model in order to assess the trustworthiness of agents in multi-agent systems. The system can calculate the expected trustworthiness of agents based on its prior experience and can also combine evidence provided by other sources in statistically sound ways. (Finally, it also offers a user interface to visualize trust when user involvement is needed.) By providing CertainTrust as a task to OpenCCE we want to enable users and devices to enhance their security in high-risk environments.

2 Implementation

We have implemented OpenCCE as a Java project. More specifically, the whole fuctionality is encompassed in a single class (CertainTrust.java). The remainder of our classes support a graphical user interface intended to visualize trust in a user-friendly way.

3 Usage examples and rules

3.1 Usage examples

The proposed use of our task consists of a CertainTrust object constructed to handle a single trust relationship. The constructor is reached by calling one of the *createfromRS* or *createfromTC* functions. After the constructor, the main and presumably more regularly used functionalities are implemented by the functions *addR*, *addS* and *addU* that respectively add positive, negative and neutral evidence to the statistical model. Furthermore, there are a number of ways to combine CertainTrust objects by choosing a function from the set {*OR, AND, wFusion, cFusion*}.

A typical example of the use of the task could be the creation of a computational trust object and then the incorporation of evidence in the statistical model that leads to an extracted expectation value for the trustworthiness of the given agent. A small snippet is found below:

```
trustObject = CertainTrust.createDefault(N, name)
trustObject.addR(posEvidence)
trustObject.addS(negEvidence)
expectation = trustObject.getExpectation()
```

3.2 Rules

Rules regarding the use of CertainTrust address the correctness of the model and are provided as comments to each function implemented in the class. They are normally handled by exceptions, but a static analysis could have some benefit.

4 Algorithms Model

CertainTrust in its current form does not offer any variability. The task, which could be named “Initialize CertainTrust Object” is self-contained in the Java class of the same name. Providing different tasks for each operation on the trust objects and grouping them together could be an option, but is not considered practical at his point.

5 Questions for initilization

The main user choices during the initialization of the task concern the arguments of the *createfromRS* and *createfromTC*. More specifically, the user should be able to choose the name of the object, an initial (prior) trust value as well as the number N of adequate evidence.