We decided that we add a method that we thought is going to be useful for the system, the feature was that we added a new method to the class JavaSource\org\unitime\commons\Debug.Java, the method is:

/\*\*

\* Logs a stack trace at the debug level.

\* @param throwable the throwable to log the stack trace for

\*/

public static synchronized void logStackTrace(Throwable throwable) {

if (throwable != null) {

debug(throwable.toString());

for (StackTraceElement element : throwable.getStackTrace()) {

debug("\tat " + element);

}

if (throwable.getCause() != null) {

debug("Caused by:");

logStackTrace(throwable.getCause());

}

}

}

This method takes a Throwable object as its parameter and logs the stack trace of the exception at the debug level. It uses the debug method already present in the Debug class to log the exception's class name and message, and then iterates over the stack trace elements to log each one using the debug method as well.

If the exception has a cause, the method recursively logs the cause's stack trace as well, with an additional message indicating that it was caused by the previous exception.

This method could be useful for debugging exceptions in a Java application, as it provides a detailed log of where the exception occurred and how it was propagated through the code.