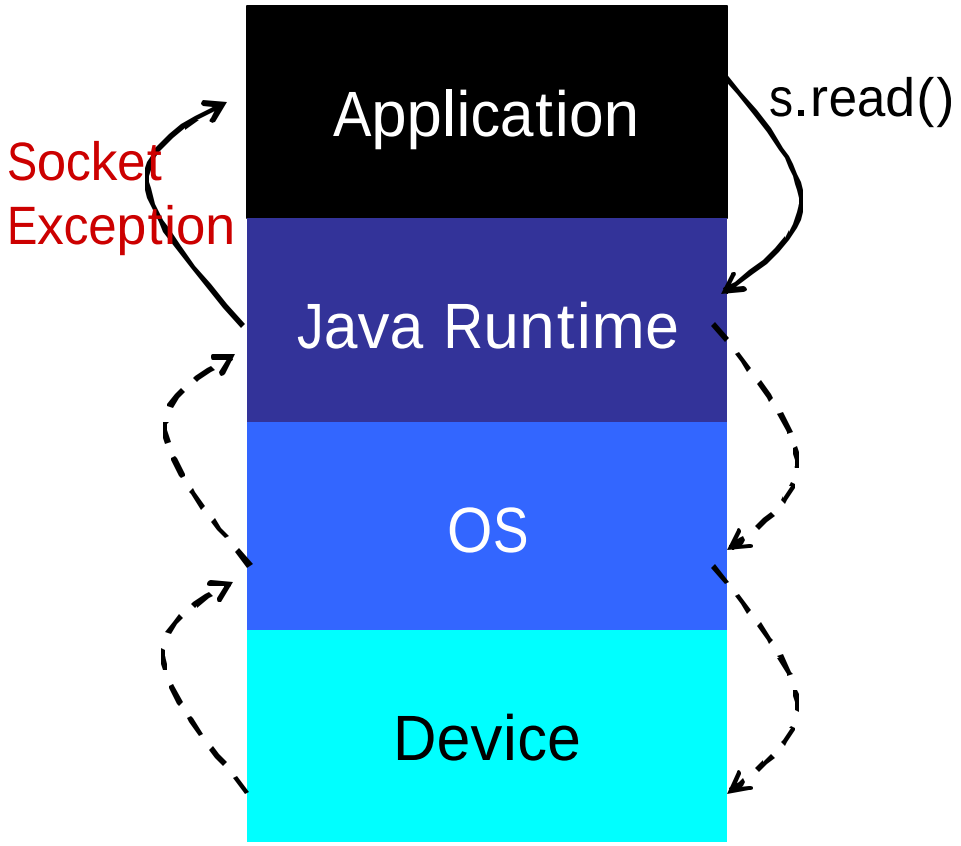
Navigating Error Recovery Code in Java Applications

Chen Fu and Barbara Ryder

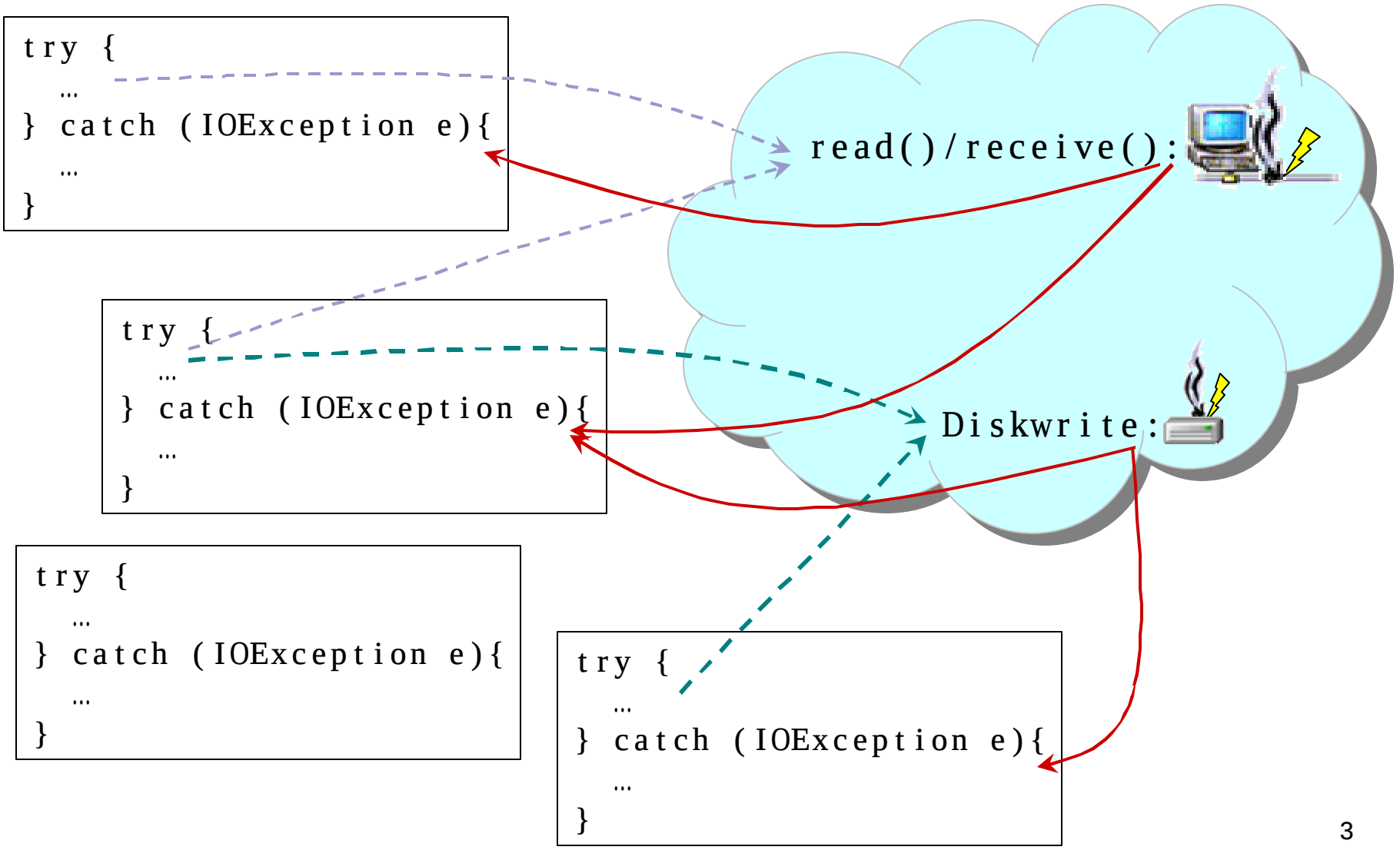
Rutgers University

Exception Handling

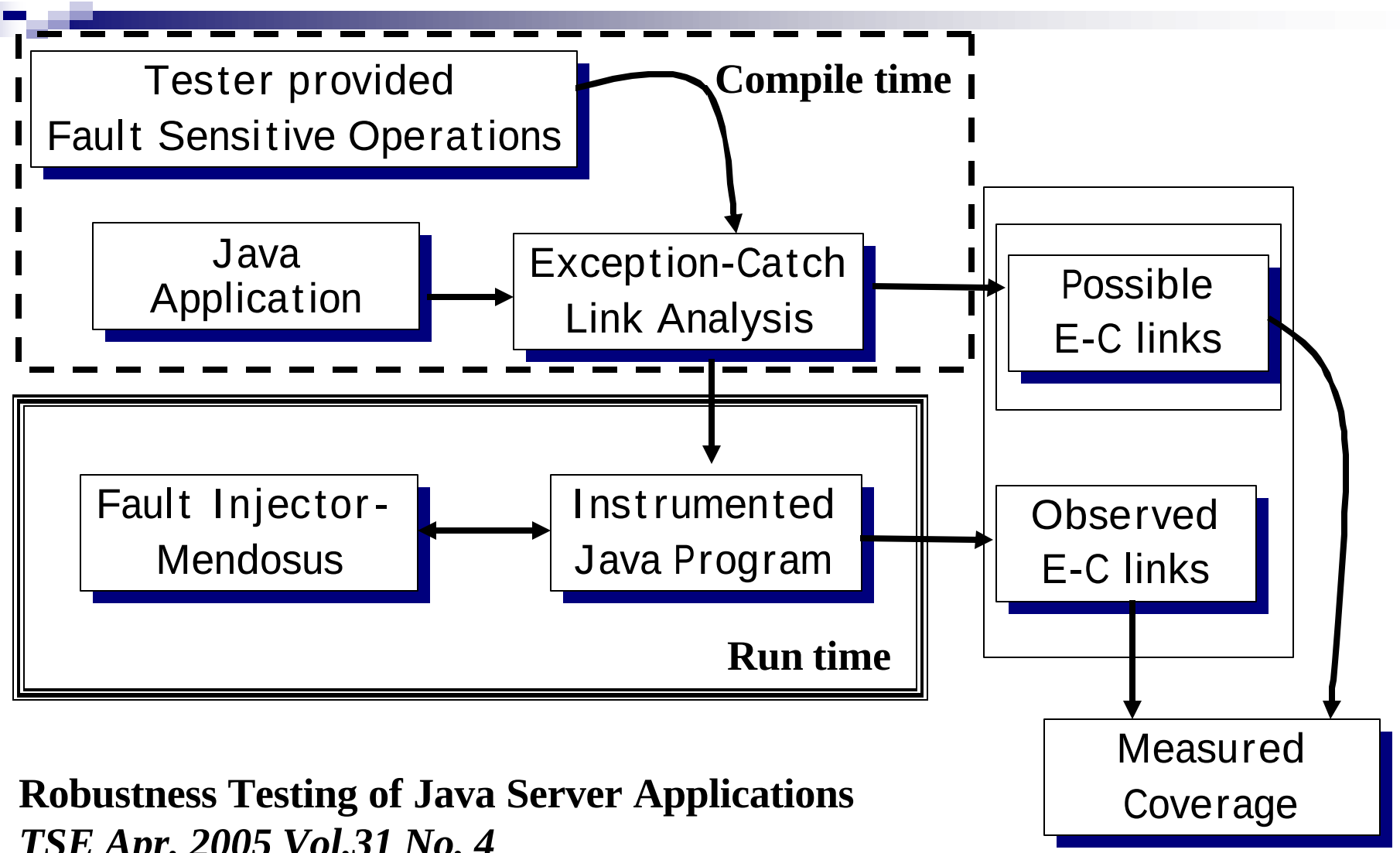
- Java Applications run on layered system
- Subsystem Faults are translated into exceptions
- Proper handling is important



Exception-Catch Links



Testing Framework

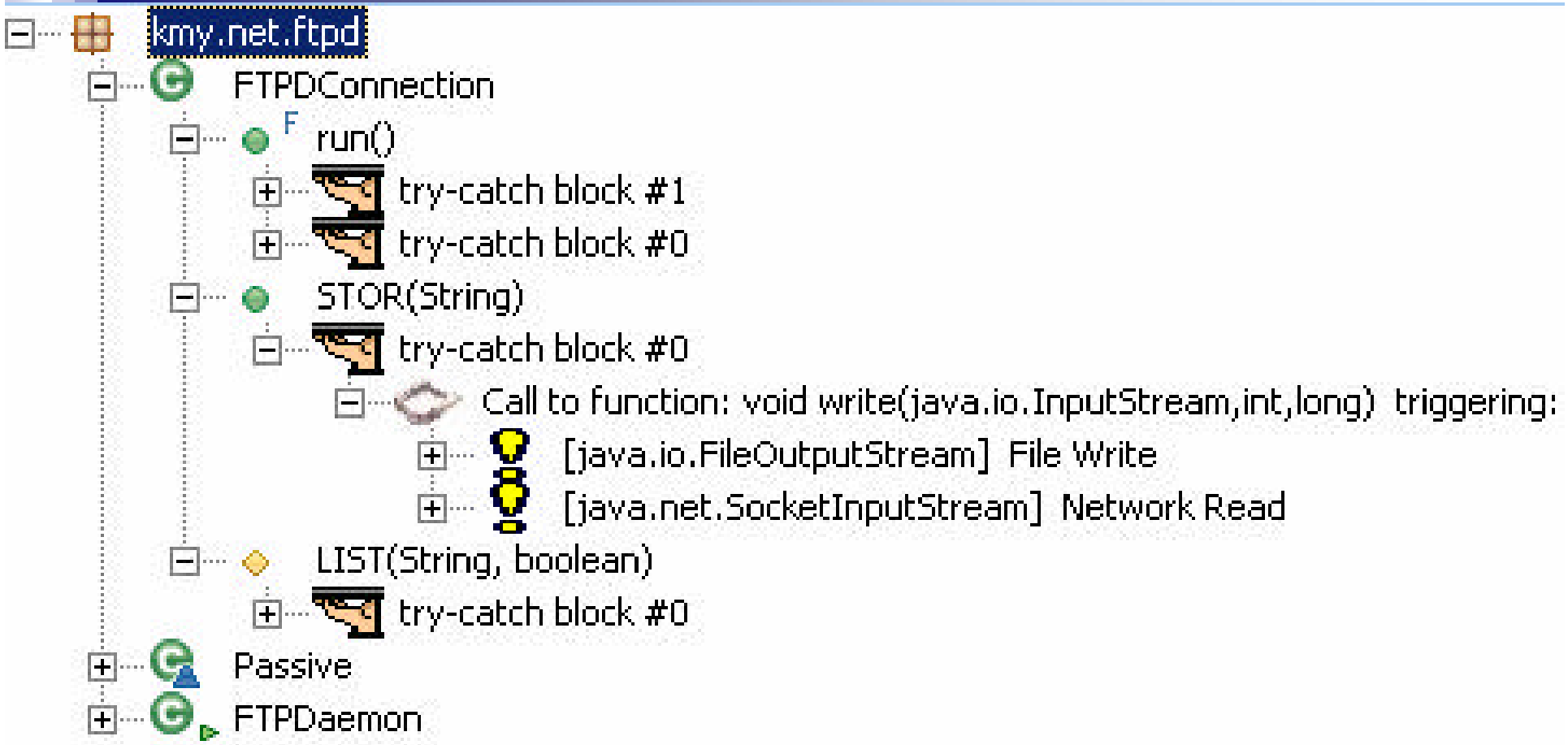


Robustness Testing of Java Server Applications
TSE Apr. 2005 Vol.31 No. 4

ExTest – Displaying *e-c links* in an IDE

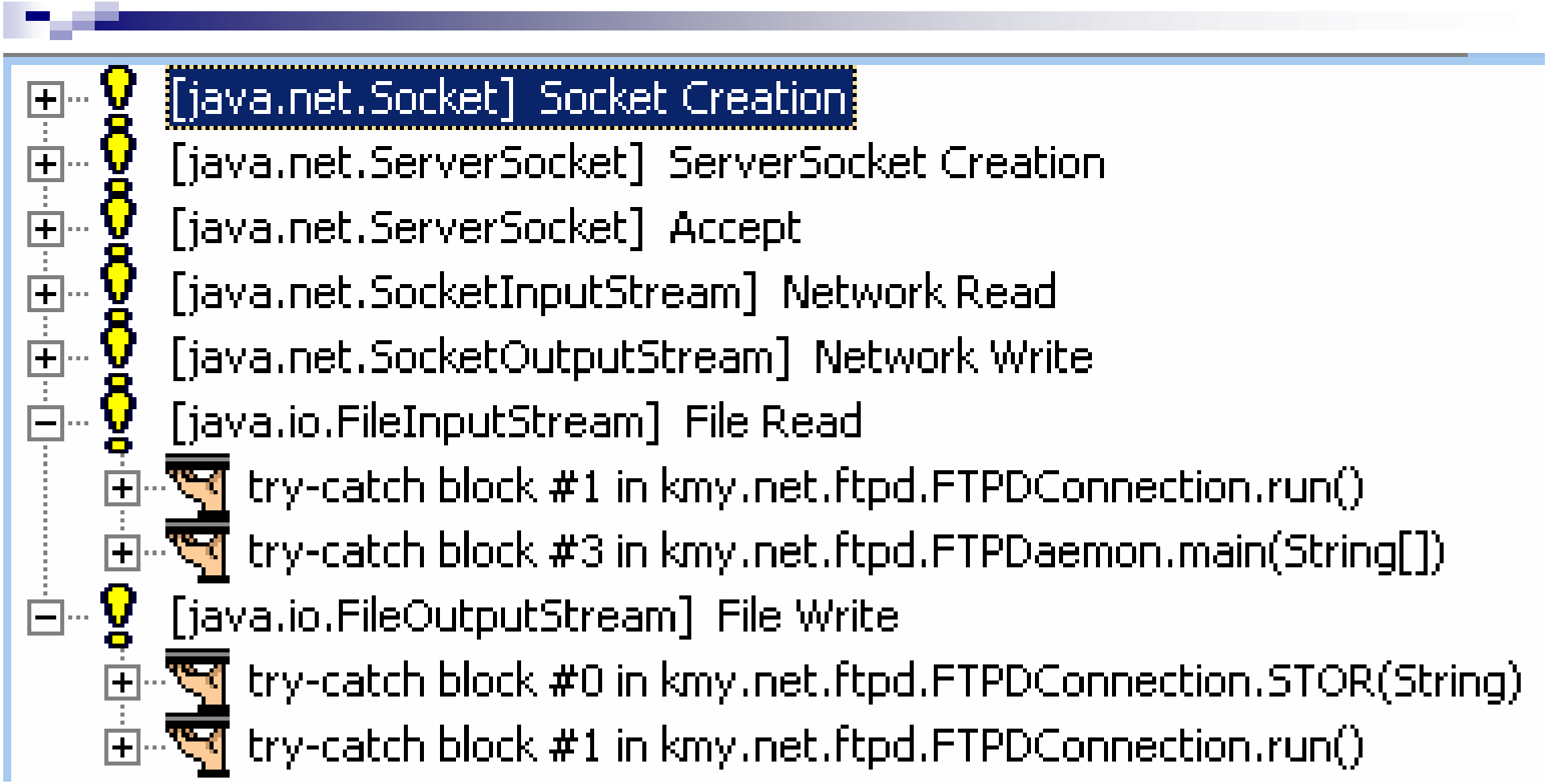
- *e-c links*
  - Propagation paths of exceptions
- Program Understanding (error recovery part)
  - Visually display exception triggering points and the corresponding handling code.
- Testing
  - Help identifying spurious e-c links

E-C links grouped by “catches”

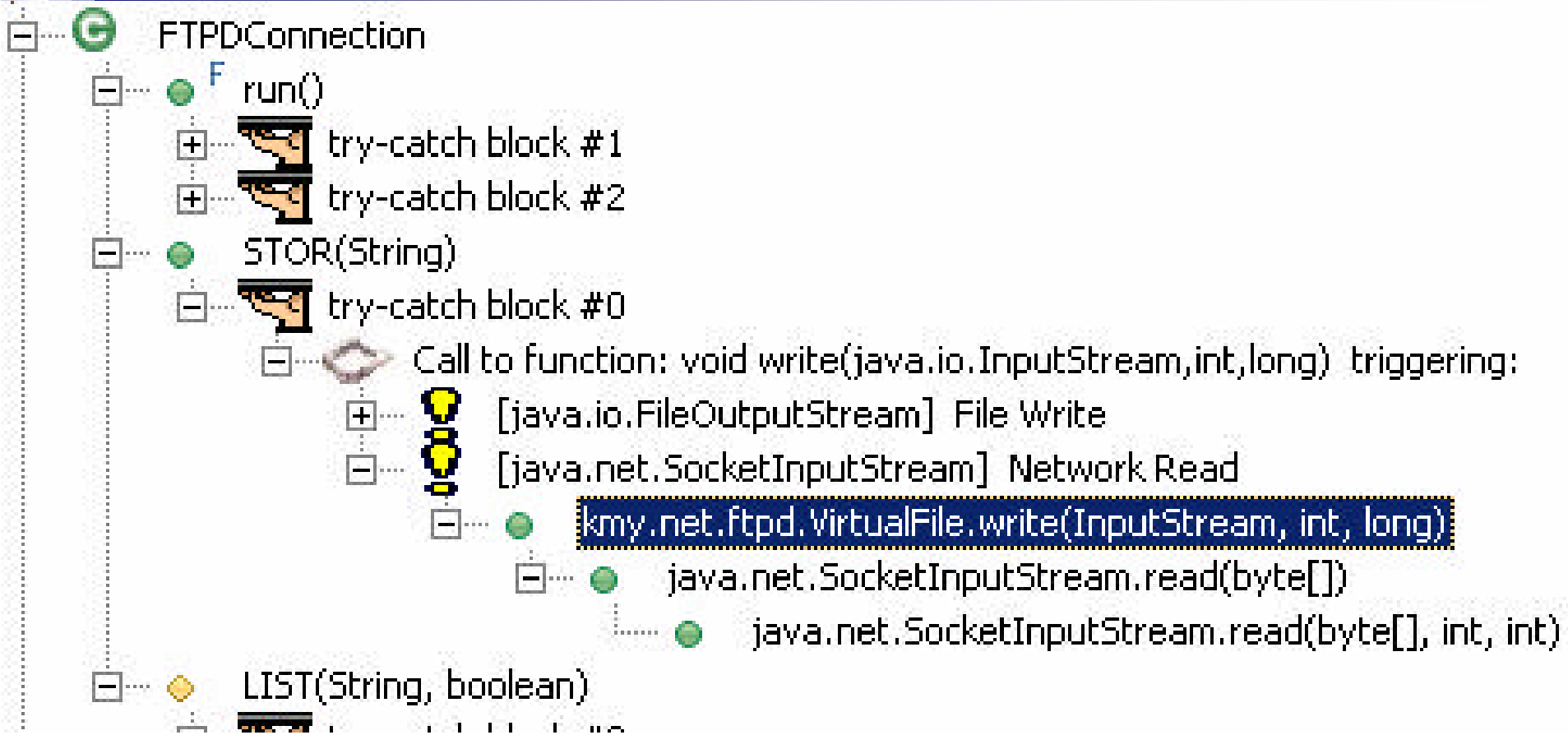


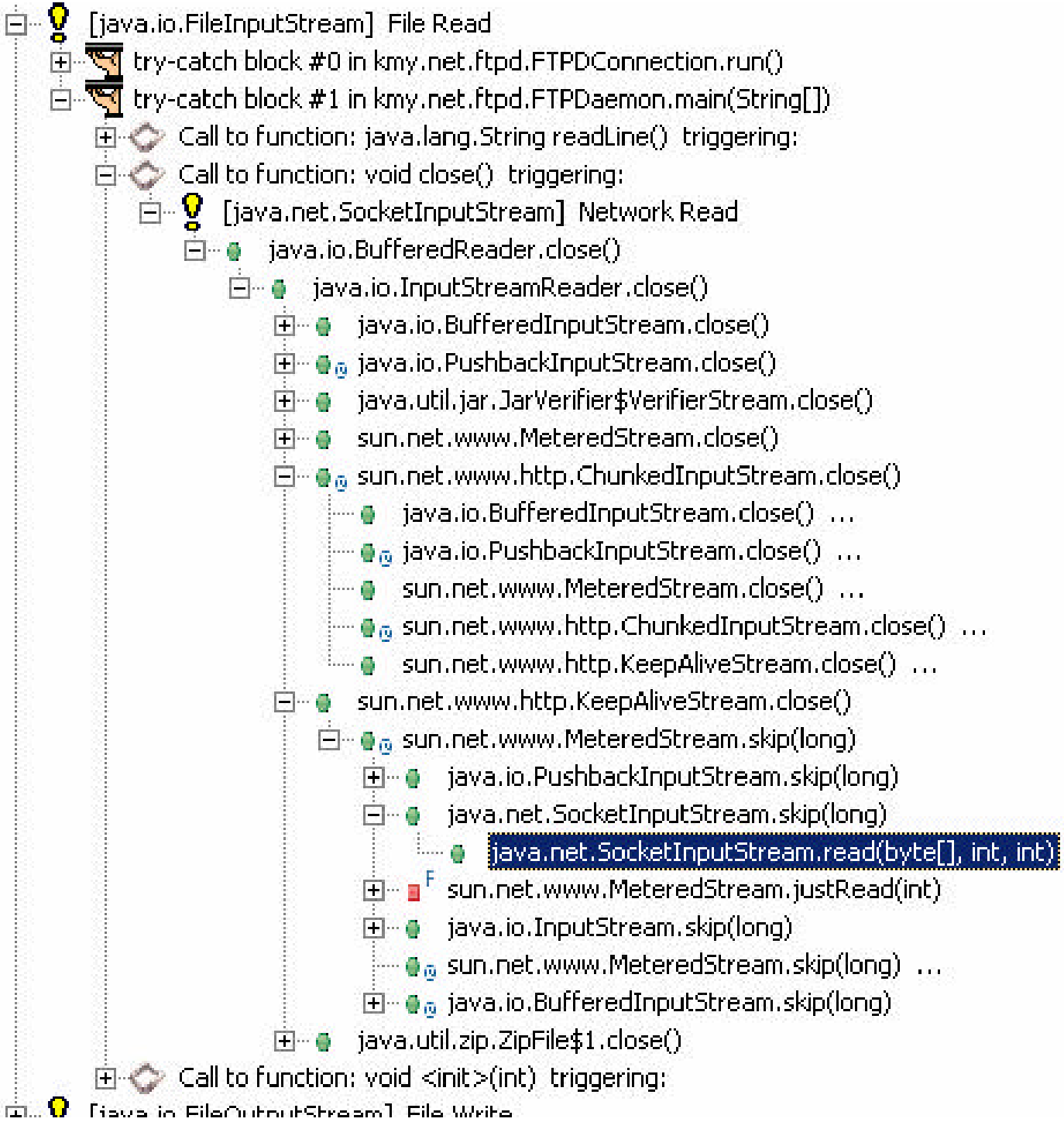
Screen shot of benchmark: FTPD in Java by Peter Sortokin.

E-C Links grouped by “triggers”



E-C Link


FTPDConnection
run()
try-catch block #1
try-catch block #2
STOR(String)
try-catch block #0
Call to function: void write(java.io.InputStream,int,long) triggering:
[java.io.FileOutputStream] File Write
[java.net.SocketInputStream] Network Read
kmy.net.ftpd.VirtualFile.write(InputStream, int, long)
java.net.SocketInputStream.read(byte[])
java.net.SocketInputStream.read(byte[], int, int)
LIST(String, boolean)


[java.io.FileInputStream] File Read
try-catch block #0 in kmy.net.ftpd.FTPDConnection.run()
try-catch block #1 in kmy.net.ftpd.FTPDaemon.main(String[])
Call to function: java.lang.String readLine() triggering:
Call to function: void close() triggering:
[java.net.SocketInputStream] Network Read
java.io.BufferedReader.close()
java.io.InputStreamReader.close()
java.io.BufferedInputStream.close()
java.io.PushbackInputStream.close()
java.util.jar.JarVerifier$VerifierStream.close()
sun.net.www.MeteredStream.close()
sun.net.www.http.ChunkedInputStream.close()
java.io.BufferedInputStream.close() ...
java.io.PushbackInputStream.close() ...
sun.net.www.MeteredStream.close() ...
sun.net.www.http.ChunkedInputStream.close() ...
sun.net.www.http.KeepAliveStream.close() ...
sun.net.www.http.KeepAliveStream.close()
sun.net.www.MeteredStream.skip(long)
java.io.PushbackInputStream.skip(long)
java.net.SocketInputStream.skip(long)
java.net.SocketInputStream.read(byte[], int, int)
sun.net.www.MeteredStream.justRead(int)
java.io.InputStream.skip(long)
sun.net.www.MeteredStream.skip(long) ...
java.io.BufferedInputStream.skip(long)
java.util.zip.ZipFile$1.close()
Call to function: void <init>(int) triggering:

Related Work

- Underlying Analysis
  - **“Robustness Testing of Java Server Applications”** ***TSE Apr. 2005 Vol.31 No. 4***
- Displaying exception propagation and handling:
  - Chang et. al: Visualization of exception propagation for java using static analysis. In (SCAM 02)
  - Robillard and Murphy: Jex tool (FSE 01)
- Displaying exception usage issues:
  - Reimer and Srinivasan: SABER (EHOOS 03)
  - Sinha et. al. (ICSE 04)