ChangeMiner

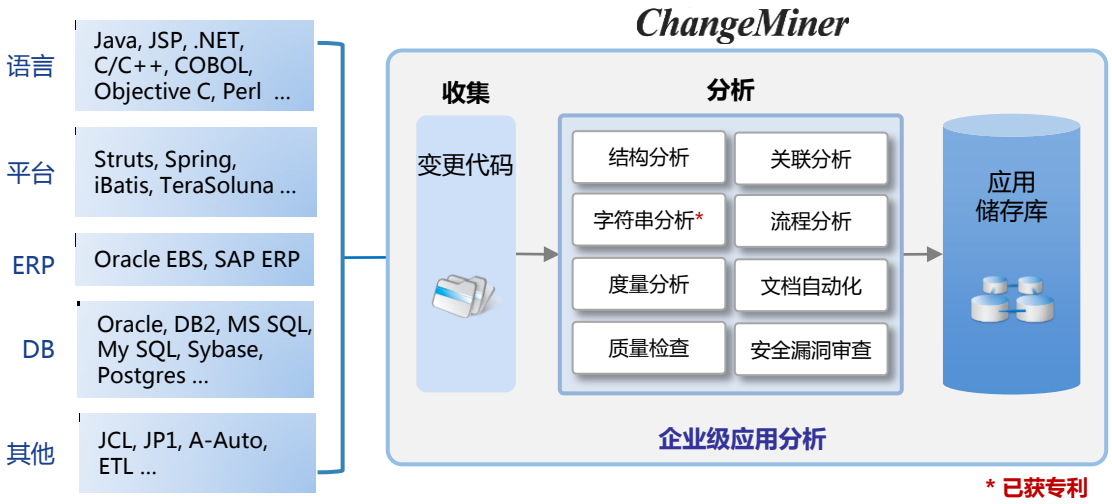
CHANGEMINER WHITE PAPER

强大的字符串分析引擎，使应用变更影响分析结果最精确

GTONE
Governance Technology

ChangeMiner是?

ChangeMiner是可视化展现企业复杂的应用以及数据库的应用自动分析工具。ChangeMiner针对多种编程语言以及技术开发的复杂的应用提供自动化知识库，从而有效提高开发及维护的生产性。

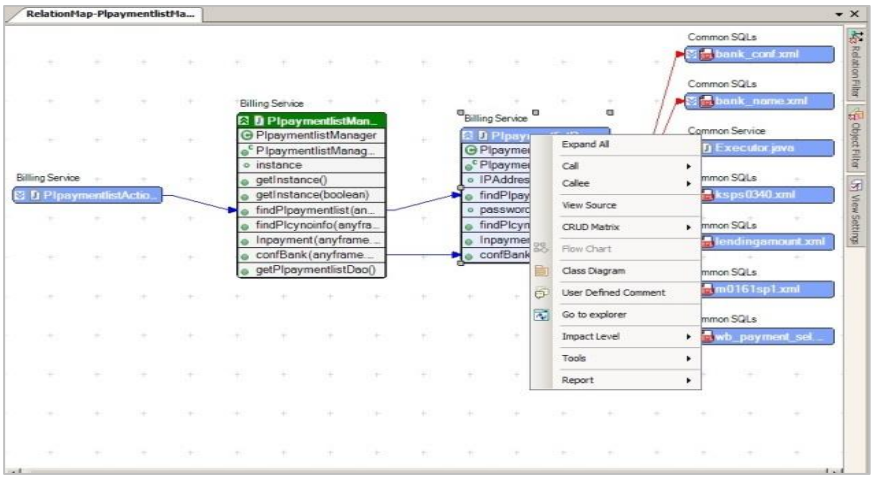


[图 1] 工具的基本概念

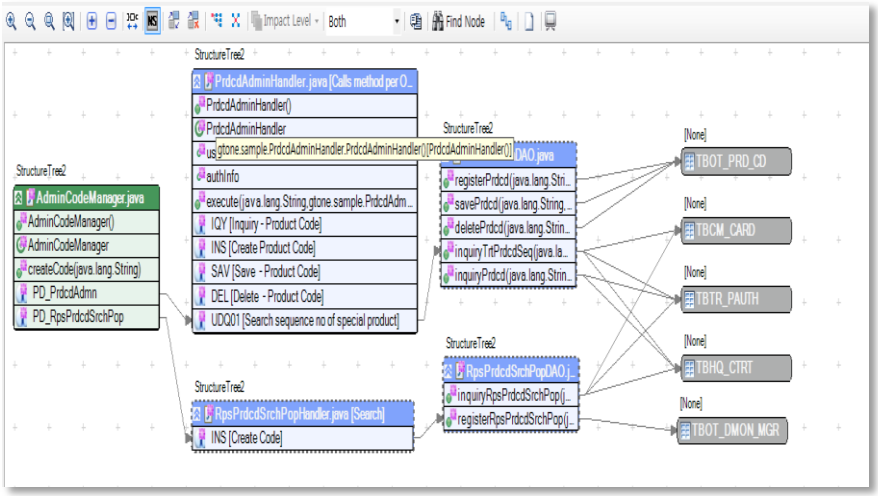
ChangeMiner所提供的信息具有强大的文档化功能。并且，以通过互联网可以随时随地访问的应用存储库进行共享。应用开发以及维护部门可以通过相应应用知识库快速地理解应用程序结构以及其关联性。

ChangeMiner主要应用方案中的之一是，进行应用程序与数据库之间的变更影响分析。例如，变更数据库Schema时，需要识别相应变更相关潜在的影响范围，从而准确修改应用程序，从而预防系统故障，ChangeMiner提供End-To-End方法调用关系信息以及CRUD Matrix[1] 等重要分析信息，使开发部门更加效率地进行变更影响分析或进行变量级别的系统故障原因追踪。

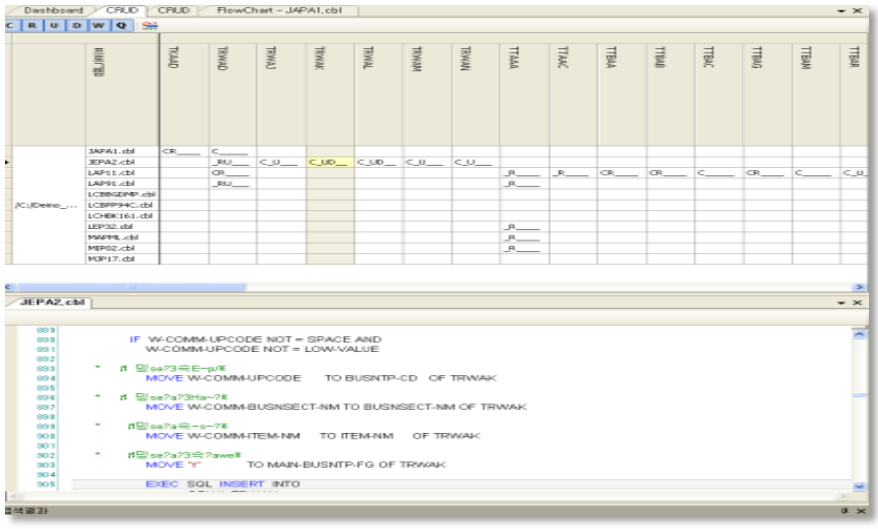
[1] CRUD (Create, Read, Update, Delete) Matrix 提供，通过何种程序、何种类型的SQL，调用数据库对象相关信息。



[图 2] End-To-End 方法调用关系例子



[图 3] 根据条件值进行方法调用的例子



[图 4] CRUD Matrix 例子

使 ChangeMiner 更加特别的要素

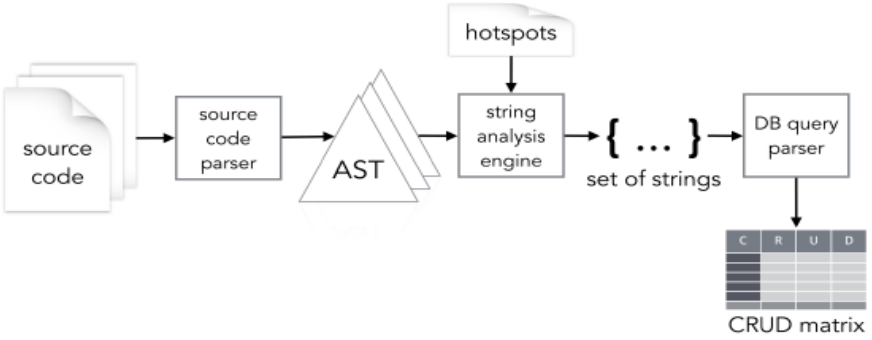
综合字符串分析器(Universal String Analyzer) 是将ChangeMiner与其他竞争产品差别化的核心技术。

对于大部分将特定字符串与用户输入值进行拼接后，通过运行时所决定的SQL语句与数据库进行通讯的应用程序，准确分析相应程序的工作是非常困难。ChangeMiner为了克服上述技术性屏障，除了使用目前广泛所使用的语法解析技术以外，同时使用综合字符串分析(专利)技术。

ChangeMiner内嵌的综合字符串分析器是为了不限于所使用的编程语言而所使用的字符串静态分析引擎。相应引擎以下述先进技术为基础，在不执行应用程序的情况下，可以准确的提取源代码内部的动态字符串。

- 通过函数间路径敏感性分析(Inter-procedural path-sensitive analysis)，从而减少False positive。
- 通过深度数据结构分析，从而确保其准确性。

图5为综合字符串分析引擎的结构。相应引擎通过处理源代码的AST(Abstract Syntax Tree)，生成包含控制流和数据流的Graph。相应Graph会被转换成简单的String Graph。并且，会运用在提取一系列查询语句字符串。



[图 5] 字符串分析器的结构

相应引擎被设计成可以进行路径敏感性分析。因此，各个字符串表达式包含是否为true branch或false branch的路径信息。将图6的源代码作为例子。从第3行到第11行，变量s1, s2, s3, s4, s5会根据条件，进行不同的字符串赋值。之后，会将其进行拼接后，赋给查询语句的变量。在第12行可能出现的查询语句的个数明确为4。假如，不进行路径敏感性分析，会提取可能组合的1,024(=45)个字符串。这样会导致产生大量false positive。并且，会浪费大量时间以及资源。

```
1  String query = “”;
2  String s1, s2, s3, s4, s5;
3  if (i > 0) {
4    s1 = “S1”; s2 = “S2”; s3 = “S3”; s4=”S4”; s5 = “S5”;
5    } else if (i > 1) {
6     s1 = “T1”; s2 = “T2”; s3 = “T3”; s4=”T4”; s5 = “T5”;
7     } else if (i > 2) {
8      s1 = “U1”; s2 = “U2”; s3 = “U3”; s4=”U4”; s5 = “U5”;
9     } else {
10    s1 = “V1”; s2 = “V2”; s3 = “V3”; s4=”V4”; s5 = “V5”;
11   }
12    query += “SELEC” + s1 + s2 + s3 + s4 + s5;
13    Target.hotspot (query);
```

[图 6] 路径敏感性分析必要性相关例子

图7为更加接近于现实的程序相关例子。在相应例子，会通过4个变量(company, dept, code, price)进行查询语句的拼接。4个变量中3个变量(company, dept, code)会根据执行路径有可能获取3个不同的值。假如，不执行路径敏感性分析的话，可能被预测的语句数为执行路径^变量个数=33 = 27个。

但是，执行路径敏感性分析的话，3个变量(company, dept, code)通过3个执行路径最终拼接成一个可能的字符串。被预测的语句个数与执行路径个数相同，并且为3。因此，有可能产生的字符串比率为9:1。这意味着，执行路径敏感性分析的结果与不执行时相比较，结果会准确9倍。

```
1  public class Product {
2    public static final int LAPTOP = 1;
3    public static Product[] getProduct(int type, int price) {
4       String query = "";
5       String company, dept, code;
6
7       switch (code) {
8       case LAPTOP:
9       case TABLET:
10          company = "electronics"
11          dept = "gadget";
12          code = type + "";
13          break;
14       case SERVER:
15          company = "electronics"
16          dept = "enterprise";
16          code = type + "";
17          break;
18       case ANALYZER:
19          company = "software"
20          dept = "rnd";
21          code = "1004";
22          break;
23       default:
24          return;
25       }
26
27       query = "SELECT FROM t_" + company + " WHERE dept='" + dept
28            + "' AND code=" + code + " AND price <" + price;
29
30       return performSelectQuery(query);
31    }
32 }
```

[图 7] 动态语句例子

在实际开发企业应用时，会使用大量变量以及执行路径。假设，使用2个执行路径和10个变量。并且，不支持路径敏感性分析的话，动态字符串提取结果可能为2^10 = 1,024。但，执行路径敏感性分析时，会准确提取两个字符串。即，相应路径比率为512:1。因此，不支持路径敏感性分析时，会额外的提取不必要的500个语句字符串。通过上述例子，可以明确的体现路径敏感分析所带来的价值。ChangeMiner所包含的字符串分析引擎支持路径敏感性分析。

再来看一个有深度的数据结构分析相关字符串分析例子。

```
1  String tables[] = new String[] {
2  "SQLP_SQL_TEXT" ,
3  "SQLP_SQL_PLAN" ,
4  "SQLP_SQL_PLAN_HIST" ,
5  "SQLP_DETECT_LINE" ,
6  "SQLP_EXECUTE_SUMMARY" ,
7  "SQLP_SQL"
8  };
9
10 Connection conn = null;
11 PreparedStatement ps = null;
12 try {
13    conn = DBUtil.getConnection(false);
14    for(int i=0; i<tables.length; i++) {
15      ps = conn.preparedStatement ("DELETE FROM " + tables[i] +
where_clause);
16    // where_clause = WHERE ANALYZE_TYPE_ID <> ?
17    …
```

[图 8] 结构体数据例子代码

假如，字符串分析引擎不能解析上述例子上的数组的话，会提取出如下不充分的字符串:

"DELETE FROM null WHERE ANALYZE_TYPE_ID <> ?"

ChangeMiner的字符串分析引擎支持静态字符串数组的分析。因此，如下所示，可以准确的提取动态拼接的语句:

"DELETE FROM SQLP_SQL_TEXT WHERE ANALYZE_TYPE_ID <> ?"

"DELETE FROM SQLP_SQL_PLAN WHERE ANALYZE_TYPE_ID <> ?"

"DELETE FROM SQLP_SQL_PLAN_HIST WHERE ANALYZE_TYPE_ID <> ?"

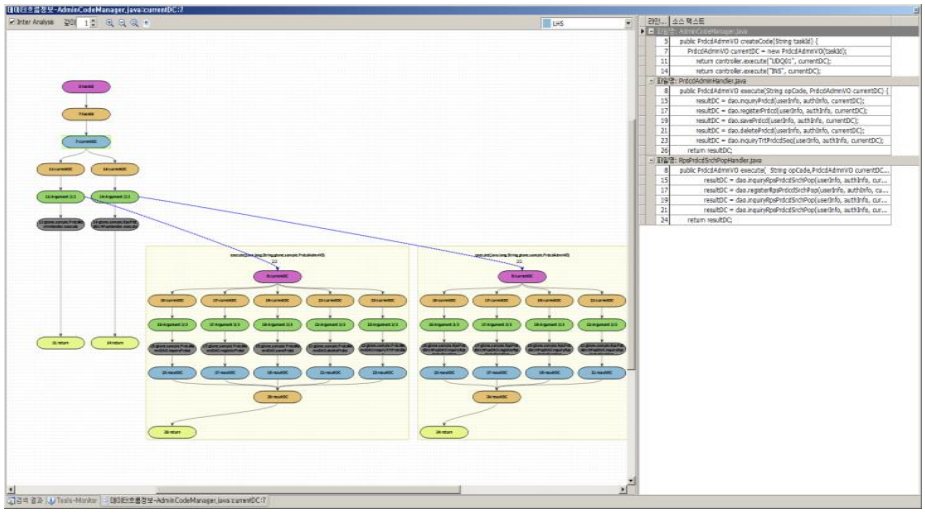
"DELETE FROM SQLP_DETECT_LINE WHERE ANALYZE_TYPE_ID <> ?"

"DELETE FROM SQLP_EXECUTE_SUMMARY WHERE ANALYZE_TYPE_ID <> ?"

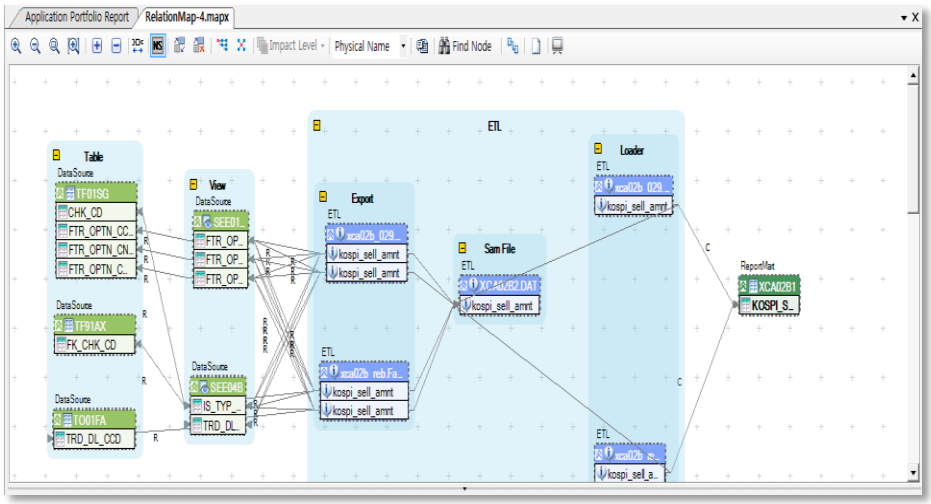
"DELETE FROM SQLP_SQL WHERE ANALYZE_TYPE_ID <> ?"

类似有深度的结构体数据的正确分析，可以使用户不会漏掉程序与数据库之间的细密的关联性。

除此之外，ChangeMiner提供，通过变量级别追踪函数间数据流分析追踪系统故障原因、为了从源代码提取业务观点信息的Structure Data Tree访问相关API、数据源分析、源代码语义静态分析等独特的功能。



[图 9] 函数间数据流分析例子



[图 10] 数据来源分析例子

结论

综合字符串分析引擎运用条件分支的路径信息，回避复杂值的分析，从而最小化false positive、确保其快速的分析性能。适用于该引擎的先进技术可以使ChangeMiner比以往更加精确的对应用程序进行变更影响分析。在多个条件重叠、条件分支分配多种字符串值、若干变量持续拼接的情况，可以更加体现出该技术的优越性。即使在不可避免false positive的情况，其性能会比不执行路径敏感性分析的情况更加卓越。

在ChangeMiner主要客户公司中，一大型IT服务企业曾报道，通过ChangeMiner的变更影响分析功能，节省了75%的时间。在使用ChangeMiner前，开发人员为了确认特定表里的字段相关影响范围，通过检索所有程序结构，并且将其文档化，整整使用了95分钟的时间。然而，导入ChangeMiner后，通过End-To-End调用关系信息、CRUD Matrix等功能，在24分钟内完成了相同的工作。

http://www.gtonesoft.com

http://www.changeminer.com

GTONE Governance Technology

Excellence in application governance