

Mining Multi-level API Usage Patterns



Mohamed
Aymen Saied



Omar Benomar



Hani Abdeen



Houari Sahraoui

DIRO, Université de Montréal, Montréal, Canada

背景

- ✧ 開発者はAPIの利用方法を，実際に利用されているコードから理解するのが一般的
- ✧ APIの利用例を探すのにコード検索エンジンサービスを用いることが多い
- ✧ コード検索エンジンサービスは，200億行のコードから検索するため，遅く重く使いにくい
- ✧ APIのメソッド利用例の検索を膨大なソースコードから容易に発見しやすくする研究が盛んにおこなわれている

早く検索できるようにしたり，検索結果の質を上げたり．．

先行研究の問題点

- ✧ API利用例の検索を支援する研究のほとんどは、1つのAPI利用例を効率的に検索するもの
- ✧ しかし、実際は他のAPIを組み合わせて用いることが多い
 - ✧ 現状では、必要なAPIの数だけ利用例を検索しなければならない
 - ✧ クエリとなるAPIと、同時に利用される別のAPIを含めた利用例を検索できるのが望ましい
 - ✧ そのためにはまず、あるAPIと同時に利用されるAPIを抽出する技術が必要

APIと同時に用いられるメソッドの例

❖ HttpClient Authentication

- HostConfigurationクラス

 - ★ setProxy(String,int)

- HttpStateクラス

 - ★ setProxyCredentials(AuthScope,Credentials)

この2つのAPIの関係は、一般的なものであるが、深く分析しないと分からない
また、ドキュメントにも載っていない

```
public class NetTools {
    private HttpClient client = null;
    ...
    public NetTools() {
        ...
        client.getHostConfiguration().setProxy( proxyHost, proxyPort );
        ...
        client.getState().setProxyCredentials( AuthScope.ANY, defaultcreds );
        ...
    }
}

public class HttpClientFactory {
    public static HttpClient getHttpClient() {
        HttpClient client = new HttpClient(new MultiThreadedHttpConnectionManager());
        ...
        HostConfiguration hc = client.getHostConfiguration();
        hc.setProxy(proxyHost, proxyPort);
        ...
        client.getState().setProxyCredentials(scope, creds);
        ...
    }
}
```

Figure 1. Code snippets of "setProxy" found using Ohloh code search engine.

目的

- ✧ あるAPIと同時に用いられるAPIの関係（APIの共利関係）マイニングする手法
(Mining Multi Level API Usage Patterns : MLUP)
を提案する

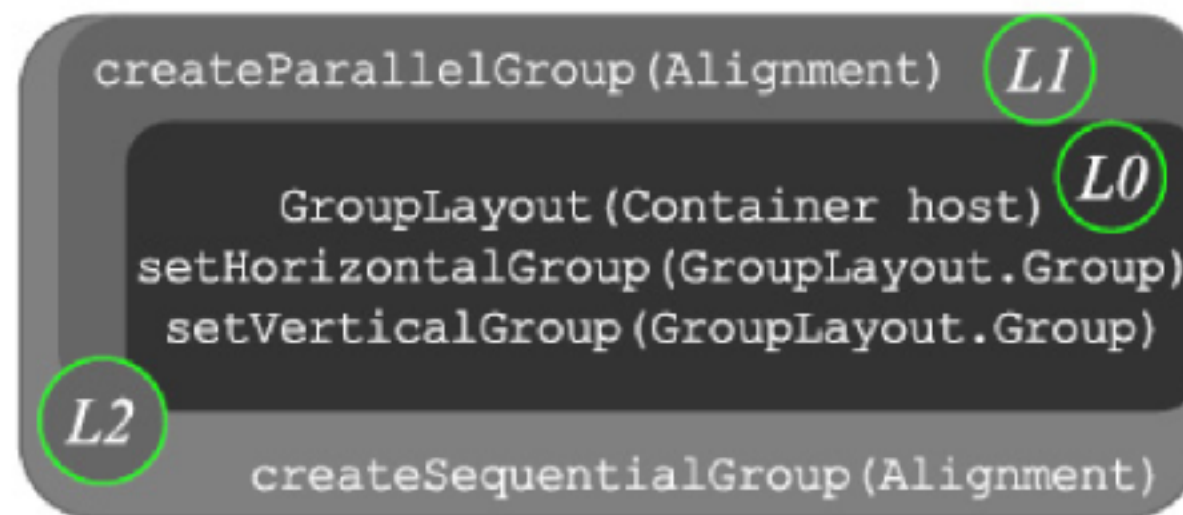


Figure 2. The cluster *L2* which represents the MLUP of class GroupLayout: *L0* represents the GroupLayout's *core* usage pattern, then the cluster *L1/L2* includes *partially/totally* the GroupLayout's *peripheral* usage pattern.

アプローチ

1. プログラムのソースコードを静的に解析して, APIとAPIを呼び出しているメソッドの共利用関係を抽出
2. APIとAPIを呼び出しているメソッドとの関係から, API呼び出しパターンをベクトルで構築
3. API呼び出しパターンからデンドログラムを作成する

API呼び出しパターン

	C1.m1	C1.m2	C2.m1	C2.m2	C2.m3	C3.m1	C3.m2
API.m1	1	1	0	0	1	1	0
API.m2	1	1	1	0	1	1	0
API.m3	0	1	1	0	1	0	0
API.m4	1	1	1	0	1	1	0
API.m5	0	0	0	1	0	0	1

Figure 3. The usage vector representation of five API methods with seven client methods.

マルチクラスタリング

＊ マルチクラスタリングというか、2つの条件でデンドログラムを作成しているだけ

- 1.クラスタの最小メソッド数

★ 値が2の場合、各クラスタに2つ以上のパターンが分類される

- 2.呼び出しパターンの類似度（ジャッカード類似度）

$$USim(m_i, m_j) = \frac{|Cl_mtd(m_i) \cap Cl_mtd(m_j)|}{|Cl_mtd(m_i) \cup Cl_mtd(m_j)|} \quad (1)$$

★ 2つのAPI呼びだしパターンの類似度が閾値以上のパターンが同じクラスタに分類

MULPによるクラスタリング

	C1.m1	C1.m2	C2.m1	C2.m2	C2.m3	C3.m1	C3.m2
API.m1	1	1	0	0	1	1	0
API.m2	1	1	1	0	1	1	0
API.m3	0	1	1	0	1	0	0
API.m4	1	1	1	0	1	1	0
API.m5	0	0	0	1	0	0	1

API.m5

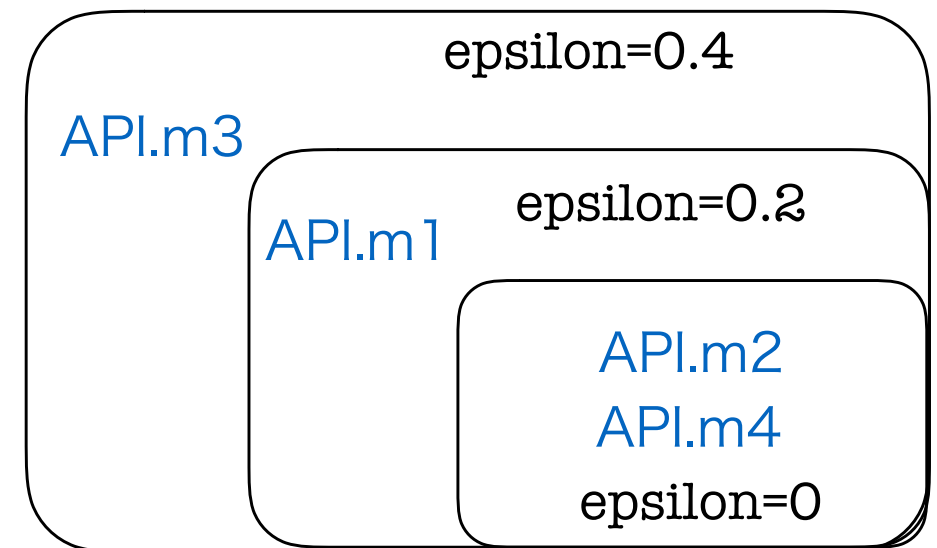


Figure 3. The usage vector representation of five API methods with seven client methods.

データセット

MULPでのクラスタリング結果

- ★ ϵ とはUsimを距離にしたもの $1 - \text{USim}(m_i, m_j)$
- ★ 同じクラスタのAPIは共利用関係にある
- ★ 次の比較実験では, $\epsilon=0.35$ で行う

比較実験

- ✧ 提案手法がAPIの共利用関係を，既存手法MAPO[1]よりどれくらい抽出できるかを検証する
 - RQ1：抽出されたAPIの共利用関係は正しいものなのか
 - RQ2：抽出されたAPIの共利用関係は，一般化できるか
- ✧ データセット
 - 4つのAPIクラス（カテゴリ）
 - ★ Httpアクセス系：HttpClient, Java Security
 - ★ 描画系：Swing, AWT

Table II
CLIENT PROGRAMS USED IN OUR CASE-STUDY

APIs	Client programs	Description
Swing & AWT	G4P (GUI for processing)	A library that provides a rich collection of 2D GUI controls
	Valkyrie RCP	A Spring port of the current Spring Rich Client codebase.
	GLIPS Graffiti editor	A cross-platform SVG graphics editor
	Mogwai Java Tools	Java 2D and 3D visual entity relationship design and modeling (ERD,SQL)
	Violet	An UML editor
	Mobile Atlas Creator	This application creates off-line raster maps
	Metawidget	A smart widget Building User Interfaces for domain objects
	Art-of-Illusion	A 3D modelling and rendering studio
	VASSAL	An engine for building and playing human-vs-human games
	Neuroph	A lightweight Java Neural Network Framework
	WoPeD	A Java-based graphical workflow process editor, simulator and analyzer
	jEdit	A mature programmer's text editor
	Spring-RCP	Provide a way to build highly-configurable, GUI-standards
	GanttProject core	An application for project management and scheduling
	Pert	The PERT plugin for GanttProject
	Htmlpdf	The html and pdf export plugin for GanttProject
	Msproject	MS-Project import/export plugin for GanttProject
	swingx	Contains extensions to the Swing GUI toolkit
	JHotDraw	A Java GUI framework for technical and structured Graphics
	RapidMiner	An integrated environment for machine learning and data mining
	Sweet Home 3D	An interior design application
	LaTeXDraw	Is a graphical drawing editor for LaTeX
HttpClient & Java Security	Jakarta Cactus	A simple test framework for unit testing server-side java code
	Apache JMeter	A project that can be used as a load testing and measure performance tool
	Heritrix	A web crawler
	HtmlUnit	A GUI-Less browser for Java programs
	OpenLaszlo	An open source platform for the development and delivery of rich Internet applications
	Mule	A lightweight enterprise service bus (ESB) and integration framework
	RSSOwl	An aggregator for RSS and Atom News feeds.
	Apache Jackrabbit	Is an open source content repository for the Java platform.
	Axis2	A core engine for Web services.
	RESTEasy	A JBoss project that provides various frameworks to build RESTful Web Services
	WildFly	An application server
	WSO2 Carbon	An SOA middleware platform

実験方法（詳細は不明）

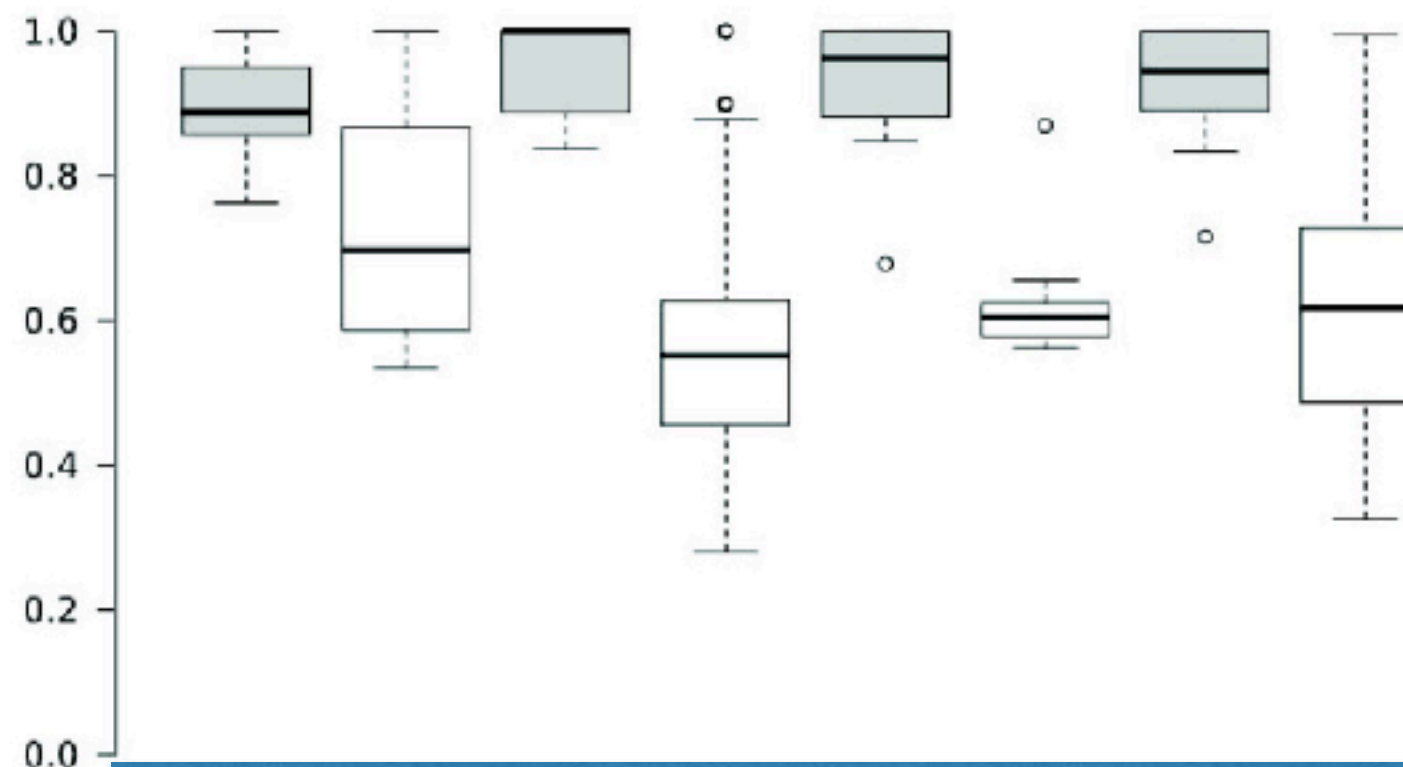
❖ 実験方法

- RQ1：抽出されたAPIの共利用関係の相関係数をカテゴリごとに調べる
- RQ2：抽出されたAPIの共利用関係がカテゴリ内で一貫して成り立つかを調べる
- leave-one-out検証
 - Swing APIを用いる22個のクライアントプログラムのうち21個を訓練データ，残りの1個をテストデータ
 - 21個で抽出されたAPIの関係は，残りの1個で成り立つか

RQ1 : 抽出されたAPIの関係は正しいものなのか

Table IV
AVERAGE COHESION OF IDENTIFIED API USAGE PATTERNS, FOR MLUP
AND MAPO.

API	MLUP	MAPO
Java Security	0.90	0.71
HttpClient	0.96	0.55
Swing	0.94	0.61
AWT	0.94	0.60



MLUPで抽出されたAPIの共利用関係の方が
相関係数が高い

Figure 1
boxes) and MAPO (white boxes).

RQ2：抽出されたAPIの関係は一般化できるか

Table VI
STATISTICS ON THE CONSISTENCY OF IDENTIFIED API USAGE PATTERNS
FOR MLUP AND MAPO

API	MLUP			MAPO		
	Avg	StdDev	Max	Avg	StdDev	Max
Java Security	0.86	0.09	0.98	0.87	0.06	0.96
HttpClient	0.77	0.09	0.89	0.86	0.03	0.90
Swing	0.83	0.05	0.92	0.91	0.02	0.95
AWT	0.84	0.04	0.94	0.77	0.03	0.88

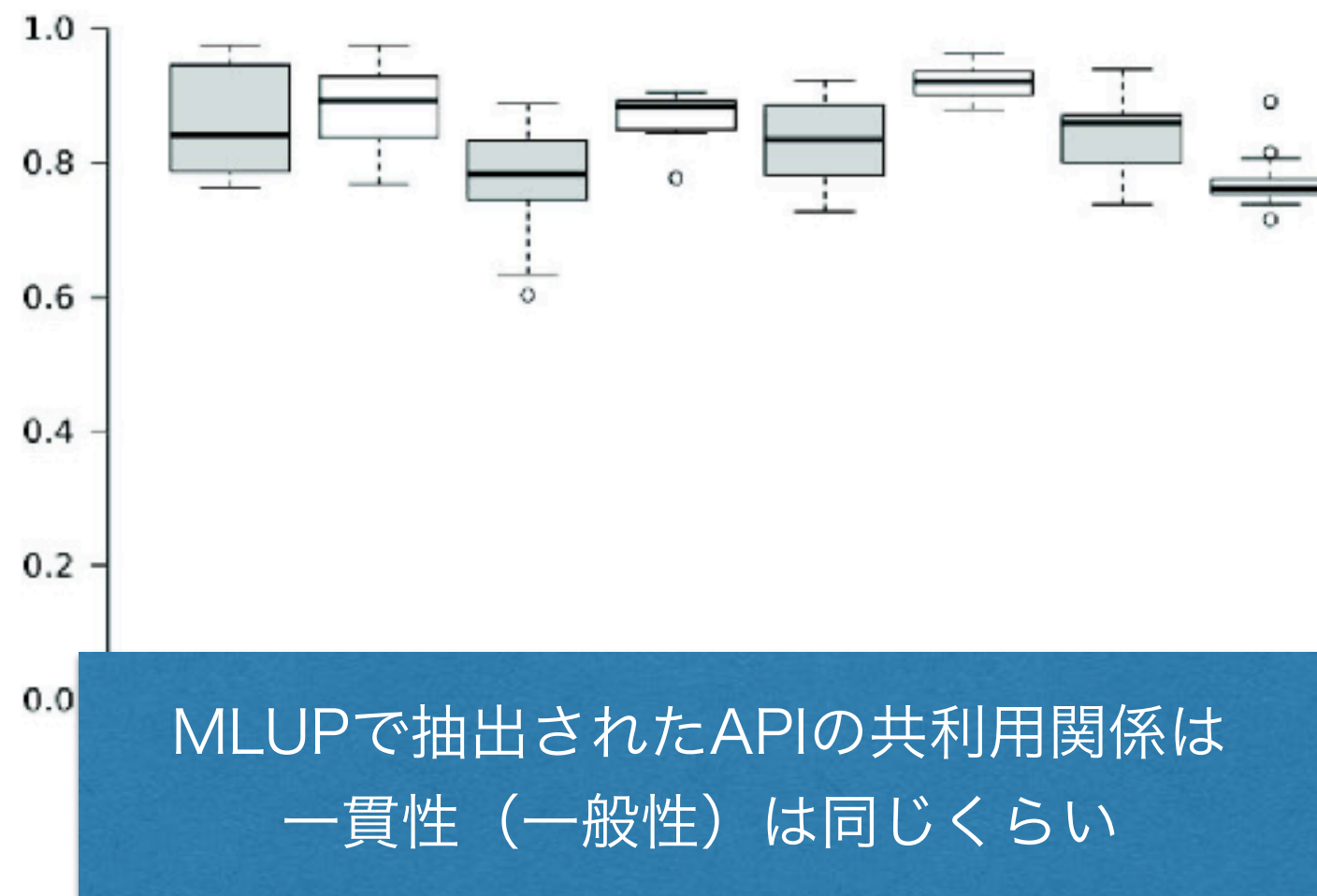


Figure 8. Consistency values of identified API usage patterns for MLUP and MAPO, across multiple validation clients.

まとめ

- ✧ 利用したいAPIと同時に用いられるAPIをマイニングする手法(Mining Multi Level API Usage Patterns : MLUP)を提案した
- ✧ 比較実験の結果
 - 既存手法であるMAPOより相関係数の高い共利用関係を抽出することができる
 - また、抽出された共利用関係の一般性は、既存手法のMAPOと同等