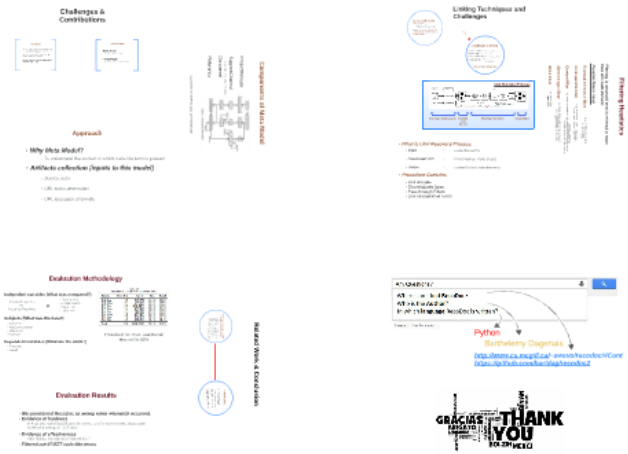
The Process Overview



UNIVERSITY OF DELAWARE

Recovering Traceability Links between an API and its Learning Resources

By,
Abhishek Kharche

UNIVERSITY OF DELAWARE

CISC879

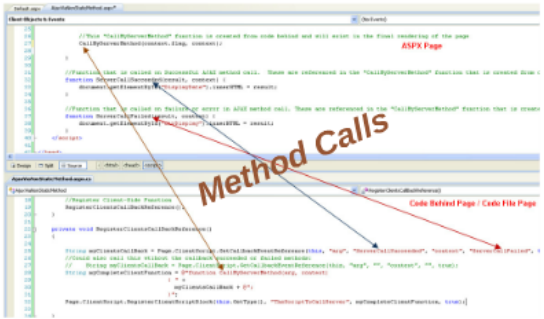
Recovering Traceability Links between an API and its Learning Resources

By,
Abhishek Kharche

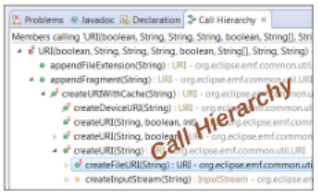
UNIVERSITY OF DELAWARE

CISC879

What is this?
Why we need it?

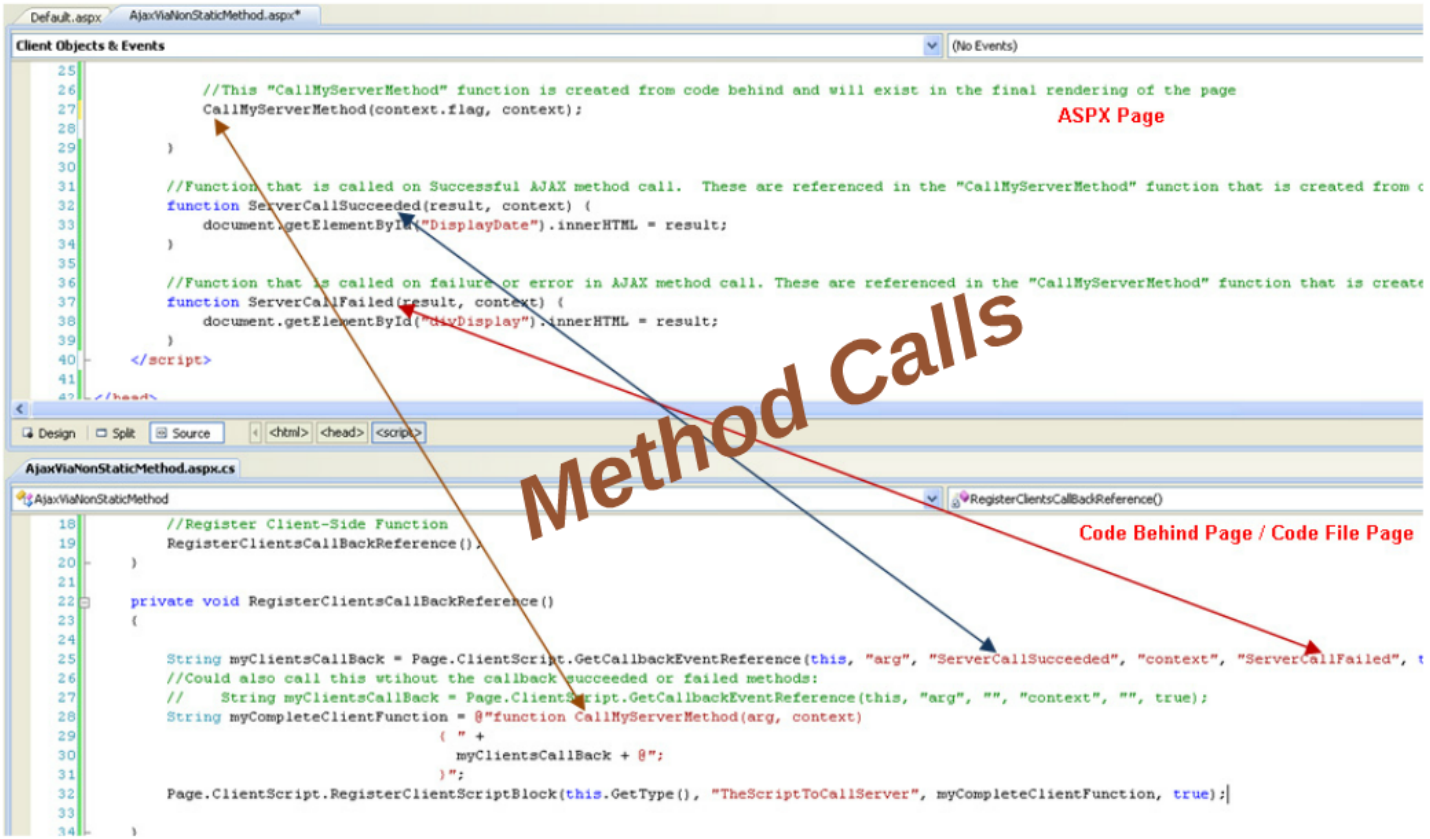


- Study of how internal components are mapped within API and resources
- Extensive efforts needed in documenting them
- When they should be updated?



So, what technique is proposed?

- Link code-like-terms (year()) to code elements (DateTime.year())
- What really is a code like term?
  e.g. patterns like () for methods
  CamelCase for types & <> for XML
- Linking is not spurious because of this process
- Implemented through tool called RecoDoc.




```
//This "CallMyServerMethod" function is created from code behind and will exist in the final rendering of the page
CallMyServerMethod(context.flag, context);

//Function that is called on Successful AJAX method call.  These are referenced in the "CallMyServerMethod" function that is created from c
function ServerCallSucceeded(result, context) {
    document.getElementById("DisplayDate").innerHTML = result;
}

//Function that is called on failure or error in AJAX method call. These are referenced in the "CallMyServerMethod" function that is create
function ServerCallFailed(result, context) {
    document.getElementById("divDisplay").innerHTML = result;
}
</script>
```

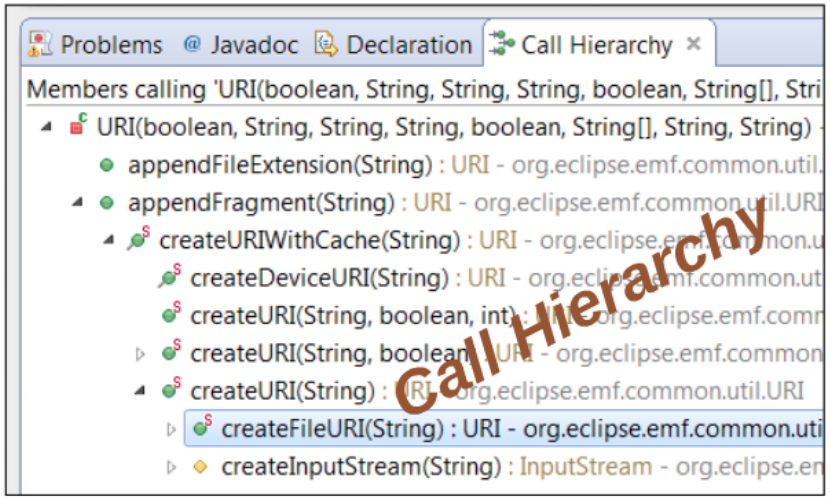
```
//Register Client-Side Function
RegisterClientsCallBackReference();
}

private void RegisterClientsCallBackReference()
{

    String myClientsCallBack = Page.ClientScript.GetCallbackEventReference(this, "arg", "ServerCallSucceeded", "context", "ServerCallFailed", t
    //Could also call this wtihout the callback succeeded or failed methods:
    //    String myClientsCallBack = Page.ClientScript.GetCallbackEventReference(this, "arg", "", "context", "", true);
    String myCompleteClientFunction = @"function CallMyServerMethod(arg, context)
                                        { " +
                                          myClientsCallBack + @";
                                        }";
    Page.ClientScript.RegisterClientScriptBlock(this.GetType(), "TheScriptToCallServer", myCompleteClientFunction, true);

}
```

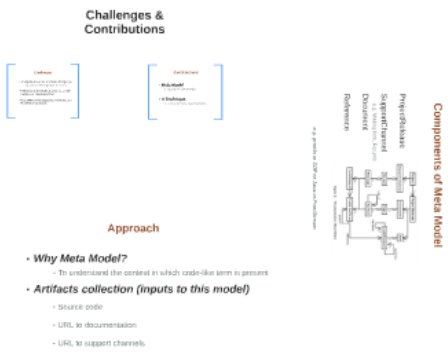
- **Study of how internal components are mapped within API and resources**
- **Extensive efforts needed in documenting them**
- **When they should be updated?**

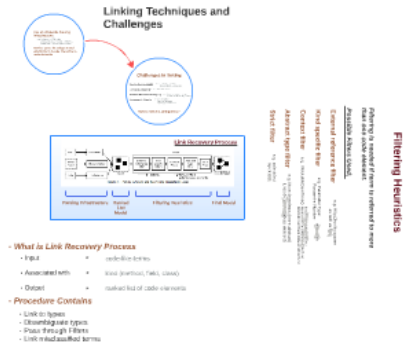


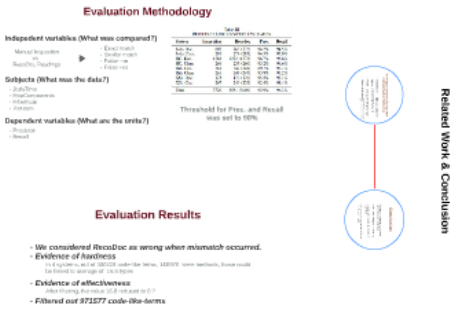
So, what technique is proposed?

- Link code-like-terms (year()) to code elements (DateTime.year())
- What really is a code like term?
  - e.g. patterns like () for methods
    CamelCase for types & <> for XML
- Linking is not spurious because of this process
- Implemented through tool called **RecoDoc.**

The Process Overview







UNIVERSITY OF DELAWARE

Challenges & Contributions

Challenges

- Ambiguity of unstructured natural language
  e.g. joda time library (year() in 11 classes)
- Reference to External Libraries, so simple mechanical match would fail
- No previously developed technique at such fine level of granularity

Contributions

- ***Meta Model***
  to represent relationships
- ***A Technique***
  to link the terms to documentation

Challenges

- Ambiguity of unstructured natural language

  e.g. joda time library (year() in 11 classes)

- Reference to External Libraries, so simple mechanical match would fail

- No previously developed technique at such fine level of granularity

Contributions

- ***Meta Model***

  to represent relationships

- ***A Technique***

  to link the terms to documentation

Approach

- ***Why Meta Model?***
  - To understand the context in which code-like term is present
- ***Artifacts collection (inputs to this model)***
  - Source code
  - URL to documentation
  - URL to support channels

Components of Meta Model

ProjectRelease

SupportChannel

e.g. Mailing lists, Forums

Document

Reference

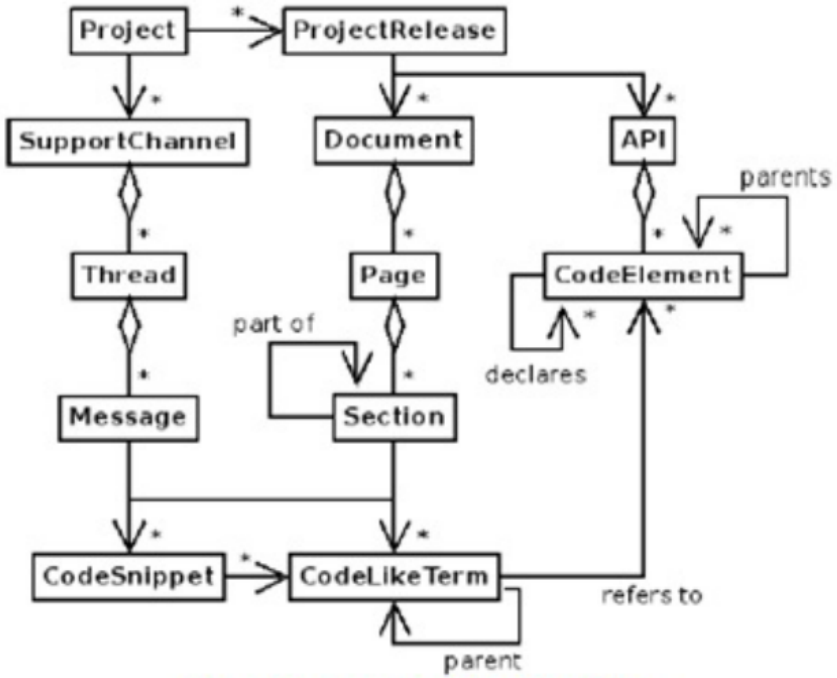


Figure 2. Documentation Meta-Model

e.g. println in SOP to Java.io.PrintStream

Linking Techniques and Challenges

Use of extensible Parsing Infrastructure

e.g. Diffference in HTML o/p of Docbook (DocBookParser) + Maven Tool (MavenParser) = DocumentationParser

Further parse the snippets and attempt to link code like terms to code elements

Challenges in linking

Declaration Ambiguity e.g. declaration of methods without their type and package

Overload Ambiguity e.g. overloaded methods

External Reference Ambiguity e.g reference to external libraries like Java Standard Library or jUnit

Language Ambiguity e.g. typo - HtttpClient
case - basiclineparser
forgetting parameters in call

How to remove ambiguities?

Link Recovery Process

Use of extensible Parsing Infrastructure

e.g. Diffference in HTML o/p of
Docbook (DocBookParser) + Maven Tool (MavenParser) = DocumentationParser

Further parse the snippets and attempt to link code like terms to code elements

Challenges in linking

Declaration Ambiguity e.g. declaration of methods without their type and package

Overload Ambiguity e.g. overloaded methods

External Reference Ambiguity e.g reference to external libraries like Java Standard Library or jUnit

Language Ambiguity e.g. typo - HtttpClient
case - basiclineparser
forgetting parameters in call

How to remove ambiguities?

How to remove ambiguities?

Link Recovery Process

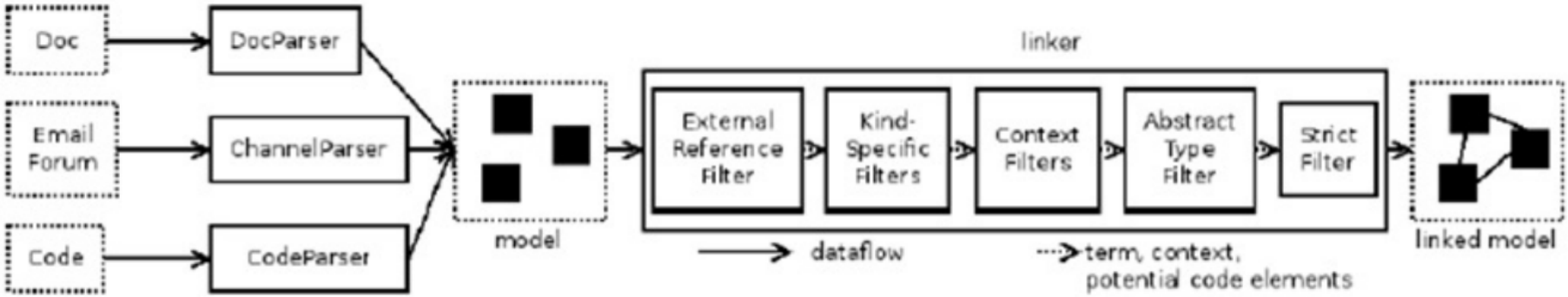


Figure 3. Parsing Artifacts and Recovering Traceability Links

Parsing Infrastructure

Ranked List Model

Filtering Heuristics

Final Model

ink Recovery Process

- ***What is Link Recovery Process***
  - Input → code-like-terms
  - Associated with → kind (method, field, class)
  - Output → ranked list of code elements
- ***Procedure Contains***
  - Link to types
  - Disambiguate types
  - Pass through Filters
  - Link misclassified terms

Filtering Heuristics

Filtering is needed if term is referred to more than one code element.

Possible Filters Used;

External reference filter e.g. HttpClient is system as well as type

Kind specific filter e.g. Parameter Type, Parameter Number mismatch

Context filter e.g. toMutableDateTime() Instant.toMutableDateTime, Abstract Instant.toMutableDateTime, ReadableDateTime.toMutableDateTime

Abstract type filter e.g. More decedents (more abstract), Less decendents(less abstract)

Strict filter e.g. select first, reject rest

Evaluation Methodology

Indepedent variables (What was compared?)

Manual Inspection
vs
RecoDoc Readings

- Exact match
- Similar match
- False –ve
- False +ve

Subjects (What was the data?)

- JodaTime
- HttpComponents
- Hibernate
- Xstream

Dependent variables (What are the units?)

- Precision
- Recall

Table III
RESULTS OF LINK RECOVERY EVALUATION

System	Inspection	Recodoc	Prec.	Recall
Joda Doc.	807	763 (772)	96.2%	94.5%
Joda Chan.	291	279 (283)	96.5%	95.9%
HC. Doc.	1288	1272 (1273)	98.7%	98.8%
HC. Chan.	266	257 (260)	95.2%	96.6%
Hib. Doc.	361	349 (349)	89.7%	96.7%
Hib. Chan.	265	247 (247)	93.9%	93.2%
XSt. Doc.	175	170 (170)	95.5%	97.1%
XSt. Cha.	267	244 (255)	92.4%	91.4%
Total	3720	3581 (3609)	95.9%	96.3%

Threshold for Prec. and Recall was set to 90%

Evaluation Results

- ***We considered RecoDoc as wrong when mismatch occurred.***
- ***Evidence of hardness***

  In 4 systems, out of 300228 code-like-terms, 160970 were methods, those could be linked to average of 16.8 types

- ***Evidence of effectiveness***

  After filtering, the value 16.8 reduced to 0.7

- ***Filtered out 971577 code-like-terms***

Related Work & Conclusion

Techniques proposed using NLP and Information Extraction

- ***Antonoil et al.*** applied VSM and probabilistic model to find pages in reference manual.
- ***Hipikat*** a tool to generate project momory from bug reports, commits, documents etc
- ***Xfinder*** a tool that matches steps of a tutorial to the code elements that implements each step
- ***Dekel and Herbsleb*** devised eMoose tool that highlights the method call in the editor when it is called

Conclusion

- Proposed a technique to link code-like-terms to documentation
- Took care of ambiguity and filtering
- In case study, Precision and Recall were 96%
- Bacchelli’s comparison found Regex are effective than IR

Any Questions?

Where can I find **RecoDoc**?
Who is the **Author**?
In which **language** RecoDoc is written?

Press Enter to search.

Python

Barthelemy Dagenais

http://www.cs.mcgill.ca/~swevo/recodoc/#Cont
https://github.com/bartdag/recodoc2