

Improving IDE Recommendations by Considering Global Implications of Existing Recommendations

Kıvanç Muşlu,[†]

Yuriy Brun,[†] Reid Holmes,[✳]

Michael D. Ernst,[†] and David Notkin[†]

[†] University of Washington

[✳] University of Waterloo

IDE recommendations

Auto Complete

```
public void setArg(String name) {  
    name_ = nam  
}
```

- name : String
- name_ : string - QFS_Demo
- Name - java.util.jar.Attributes
- Name - javax.lang.model.element
- Name - javax.naming
- Name - javax.xml.soap

Press '^Space' to show Template Proposals

Quick Fix

```
public void setArg(String name) {  
    name_ = name;  
}
```

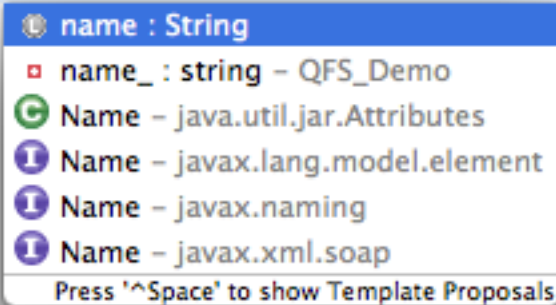
- Create class 'name_'
- Create interface 'name_'
- Change to 'Name' (java.util.jar.Attributes)
- Change to 'Name' (javax.lang.model.element)

Press 'Alt' to go to original position

IDE recommendations

Auto Complete

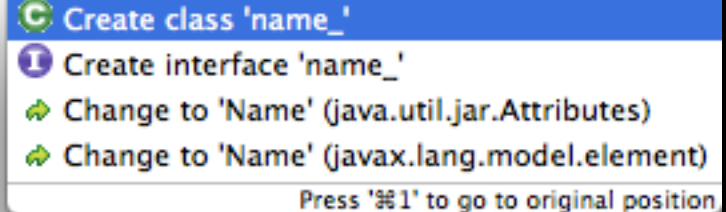
```
public void setArg(String name) {  
    name_ = nam  
}
```



The screenshot shows an IDE window with a code editor. The code is `public void setArg(String name) { name_ = nam`. The text `name_ = nam` is highlighted in blue. A dropdown menu is open below the code, listing several suggestions. The first suggestion is `name : String` with a blue background. Below it are `name_ : string - QFS_Demo`, `Name - java.util.jar.Attributes`, `Name - javax.lang.model.element`, `Name - javax.naming`, and `Name - javax.xml.soap`. At the bottom of the dropdown, it says "Press '^Space' to show Template Proposals".

Quick Fix

```
public void setArg(String name) {  
    name_ = name;  
}
```



The screenshot shows an IDE window with a code editor. The code is `public void setArg(String name) { name_ = name;`. The text `name_ = name;` is highlighted in blue. A dropdown menu is open below the code, listing several suggestions. The first suggestion is `Create class 'name_'` with a blue background. Below it are `Create interface 'name_'`, `Change to 'Name' (java.util.jar.Attributes)`, and `Change to 'Name' (javax.lang.model.element)`. At the bottom of the dropdown, it says "Press 'Esc' to go to original position".

- **Problem:** IDEs generate recommendations using local context
- **Today's contribution:** global context can make existing recommendations more powerful

Quick Fix works locally

```
public class QFS_Demo {  
    private string name_  
  
    public void setArg(String name)  
    {  
        name = name;  
    }  
}
```

Logical (root) error

Dependent error

Quick Fix works locally

```
public class QFS_Demo {  
  
    private string name_ ;  
  
    public void setArg(String name)  
    {  
        name = name;  
    }  
}
```

- 🟢 Create class 'string'
- 🟣 Create interface 'string'
- 👉 Change to 'Spring' (javax.swing)
- 👉 Change to 'String' (java.lang)
- 👉 Change to 'STRING' (javax.print.DocFlavor)
- 👉 Change to 'StringBuffer' (java.lang)
- 👉 Change to 'StringHolder' (org.omg.CORBA)
- 👉 Change to 'StringReader' (java.io)
- 👉 Change to 'StringWriter' (java.io)
- 🟡 Create enum 'string'

- 🟢 Create class 'name_'
- 🟣 Create interface 'name_'
- 👉 Change to 'Name' (java.util.jar.Attributes)
- 👉 Change to 'Name' (javax.lang.model.element)
- 👉 Change to 'Name' (javax.naming)
- 👉 Change to 'Name' (javax.xml.soap)
- 👉 Change to 'NameList' (org.w3c.dom)
- 👉 Change to 'Naming' (java.rmi)
- 🟡 Create enum 'name_'

- Two sets of recommendations
- QF for root error includes **desired** recommendation
- QF for dependent error does **not**

Quick Fix works locally

```
public class QFS_Demo {  
  
    private string name_ ;  
  
    public void setArg(String name)  
    {  
        name = name;  
    }  
}
```

- ⓐ Create class 'string'
- ⓑ Create interface 'string'
- Change to 'Spring' (javax.swing)
- Change to 'String' (java.lang)
- Change to 'STRING' (javax.print.DocFlavor)
- Change to 'StringBuffer' (java.lang)
- Change to 'StringHolder' (org.omg.CORBA)
- Change to 'StringReader' (java.io)
- Change to 'StringWriter' (java.io)
- ⓔ Create enum 'string'

- ⓐ Create class 'name_'
- ⓑ Create interface 'name_'
- Change to 'Name' (java.util.jar.Attributes)
- Change to 'Name' (javax.lang.model.element)
- Change to 'Name' (javax.naming)
- Change to 'Name' (javax.xml.soap)
- Change to 'NameList' (org.w3c.dom)
- Change to 'Naming' (java.rmi)
- ⓔ Create enum 'name_'

- Two sets of recommendations
- QF for root error includes **desired** recommendation
- QF for dependent error does **not**

Problem with IDE recommendations

- Recommendations are **local**
- Resolving some errors requires **global** info

⇒ Improve recommendations by

- Considering non-local recommendations
- Sorting local recommendations with respect to global implications

Computing global implications of Quick Fix proposals

The image shows a code editor with the following code snippet:

```
public class QFS_Demo {  
    private string name_  
  
    public void setArg(String name)  
    {  
        name = name;  
    }  
}
```

Two Quick Fix proposals are shown:

- For 'string':**
 - Create class 'string'
 - Create interface 'string'
 - Change to 'Spring' (javax.swing)
 - Change to 'String' (java.lang)
 - Change to 'STRING' (javax.print.DocFlavor)
 - Change to 'StringBuffer' (java.lang)
 - Change to 'StringHolder' (org.omg.CORBA)
 - Change to 'StringReader' (java.io)
 - Change to 'StringWriter' (java.io)
 - Create enum 'string'
 - Add type parameter 'string' to 'QFS_Demo'
 - Fix project setup...
- For 'name_':**
 - Create class 'name_'
 - Create interface 'name_'
 - Change to 'Name' (java.util.jar.Attributes)
 - Change to 'Name' (javax.lang.model.element)
 - Change to 'Name' (javax.naming)
 - Change to 'Name' (javax.xml.soap)
 - Change to 'NameList' (org.w3c.dom)
 - Change to 'Naming' (java.rmi)
 - Create enum 'name_'
 - Add type parameter 'name_' to 'QFS_Demo'
 - Rename in file (⌘2 R)
 - Rename in workspace (⌘⌘R)
 - Add type parameter 'name_' to 'setArg(String)'
 - Fix project setup...

At the bottom of the right-hand list, it says: "Press '⌘1' to go to original position".

Computing global implications of Quick Fix proposals

The image shows a code editor with the following code snippet:

```
public class QFS_Demo {  
    private string name;  
    public void setArg(String name)  
    {  
        name = name;  
    }  
}
```

Two Quick Fix proposals are shown:

- Left Panel (for 'name_'):**
 - Create class 'name_'
 - Create interface 'name_'
 - Change to 'Name' (java.util.jar.Attributes)
 - Change to 'Name' (javax.lang.model.element)
 - Change to 'Name' (javax.naming)
 - Change to 'Name' (javax.xml.soap)
 - Change to 'NameList' (org.w3c.dom)
 - Change to 'Naming' (java.rmi)
 - Create enum 'name_'
 - Add type parameter 'name_' to 'QFS_Demo'
 - Rename in file (⌘2 R)
 - Rename in workspace (⌘⌘R)
 - Add type parameter 'name_' to 'setArg(String)'
 - Fix project setup...
- Right Panel (for 'string'):**
 - Create class 'string'
 - Create interface 'string'
 - Change to 'String' (javax.swing) (highlighted in green)
 - Change to 'String' (java.lang) (highlighted in green)
 - Change to 'STRING' (javax.print.DocFlavor)
 - Change to 'StringBuffer' (java.lang)
 - Change to 'StringHolder' (org.omg.CORBA)
 - Change to 'StringReader' (java.io)
 - Change to 'StringWriter' (java.io)
 - Create enum 'string'
 - Add type parameter 'string' to 'QFS_Demo'
 - Fix project setup...

Red arrows point from the Quick Fix lists to the corresponding code elements: 'string' in the field declaration and 'name' in the assignment statement.

Press '⌘1' to go to original position

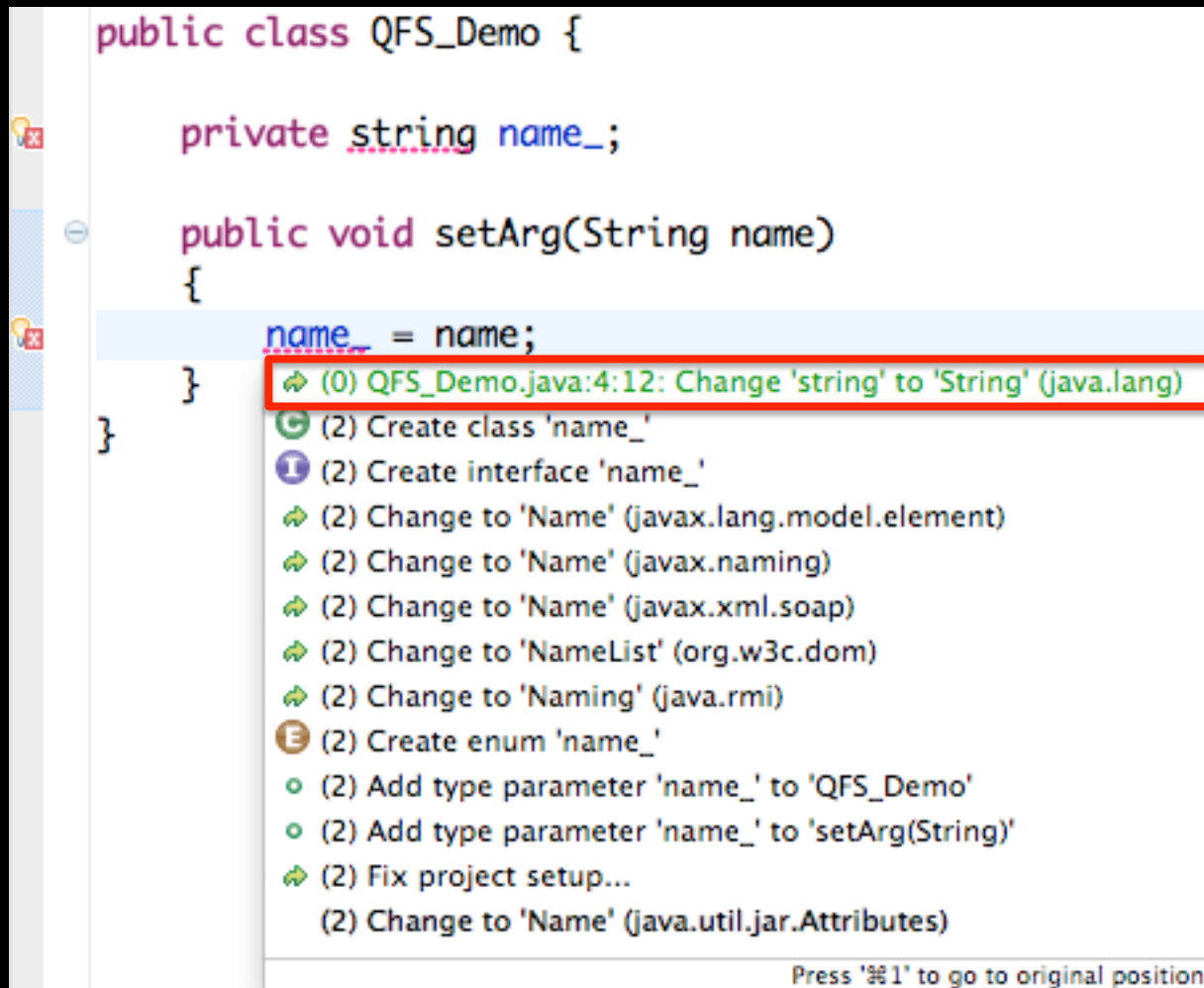
Computing global implications of Quick Fix proposals

```
public class QFS_Demo {  
    private string name_  
    public void setArg(String name)  
    {  
        name = name_  
    }  
}
```

- Change to 'String' (java.lang)
- Create class 'name_'
- Create interface 'name_'
- Change to 'Name' (java.util.jar.Attributes)
- Change to 'Name' (javax.lang.model.element)
- Change to 'Name' (javax.naming)
- Change to 'Name' (javax.xml.soap)
- Change to 'NameList' (org.w3c.dom)
- Change to 'Naming' (java.rmi)
- Create enum 'name_'
- Add type parameter 'name_' to 'QFS_Demo'
- Rename in file (⌘R)
- Rename in workspace (⇧⌘R)
- Add type parameter 'name_' to 'setArg(String)'
- Fix project setup...

Quick Fix Scout offers the Quick Fix that resolves the logical error

```
public class QFS_Demo {  
  
    private string name_  
  
    public void setArg(String name)  
    {  
        name_ = name;  
    }  
}
```



- ➡ (0) QFS_Demo.java:4:12: Change 'string' to 'String' (java.lang)
- ➡ (2) Create class 'name_'
- ➡ (2) Create interface 'name_'
- ➡ (2) Change to 'Name' (javax.lang.model.element)
- ➡ (2) Change to 'Name' (javax.naming)
- ➡ (2) Change to 'Name' (javax.xml.soap)
- ➡ (2) Change to 'NameList' (org.w3c.dom)
- ➡ (2) Change to 'Naming' (java.rmi)
- ⓔ (2) Create enum 'name_'
- (2) Add type parameter 'name_' to 'QFS_Demo'
- (2) Add type parameter 'name_' to 'setArg(String)'
- ➡ (2) Fix project setup...
- ➡ (2) Change to 'Name' (java.util.jar.Attributes)

Press '⌘1' to go to original position

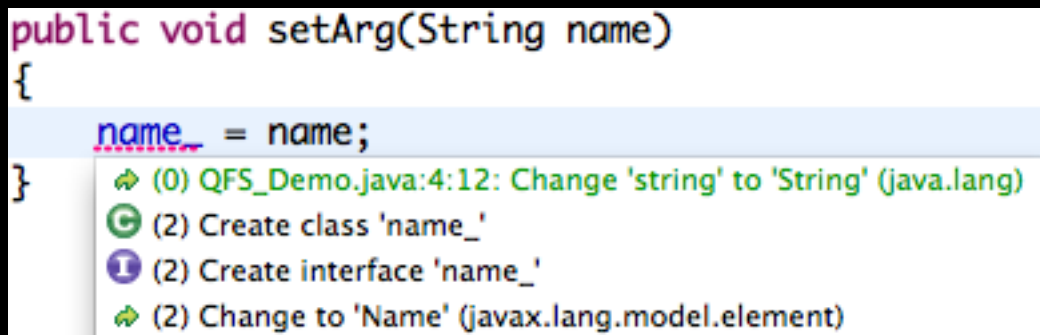
Future work

- Dependent errors
 - frequency
 - developer preference for proposals with global implications?
- Do ranking and proposals with global implications increase developer speed?
- Extend our technique to other IDE recommendations

Contributions

- Local recommendations do not always resolve dependent errors
- Considering global implications can improve recommendations
- Our technique improves as the IDE recommendations improve

```
public void setArg(String name)
{
    name_ = name;
}
```



- ➔ (0) QFS_Demo.java:4:12: Change 'string' to 'String' (java.lang)
- ⊕ (2) Create class 'name_'
- ⓘ (2) Create interface 'name_'
- ➔ (2) Change to 'Name' (javax.lang.model.element)

<http://www.quick-fix-scout.googlecode.com>

References

G. C. Murphy, M. Kersten, and L. Findlater. How are java software developers using the eclipse ide? *IEEE Software*, 23 (4):76–83, July 2006. doi: 10.1109/MS.2006.105.

E. T. Irons, “An error-correcting parse algorithm,” *Communications of the ACM*, vol. 6, pp. 669–673, 1963.

A. B. Pai and R. B. Kieburtz, “Global context recovery: A new strategy for syntactic error recovery by table-drive parsers,” *ACM TOPLAS*, vol. 2, pp. 18–41, 1980.

B. S. Lerner, M. Flower, D. Grossman, and C. Chambers, “Searching for type-error messages,” in *PLDI*, San Diego, CA, USA, June 2007, pp. 425–434.