

Exploring the Use of **Concern Element Role Information** in Feature Location Evaluation

Emily Hill, David Shepherd, Lori Pollock



Motivation:

Differentiate Element Roles in Features



Increase human annotator agreement
in creating gold sets



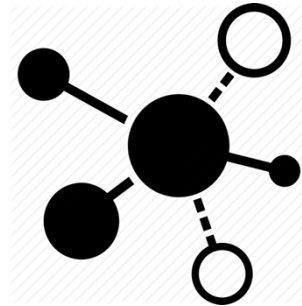
New insights into tool results
with different categories

Concern Element Roles



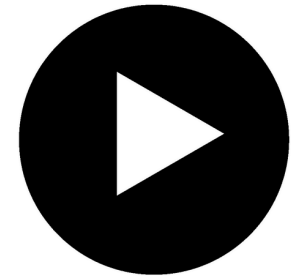
Trigger

- Initiate action (UI)



Connector

- Data preparation
- Message passing (control)



Action

- Implement action or call library



- **Initializer:** enable trigger/action (optional)
 - e.g., register UI or event listeners
- **Data:** pass data into/out of concern
 - e.g., getter/setter, field

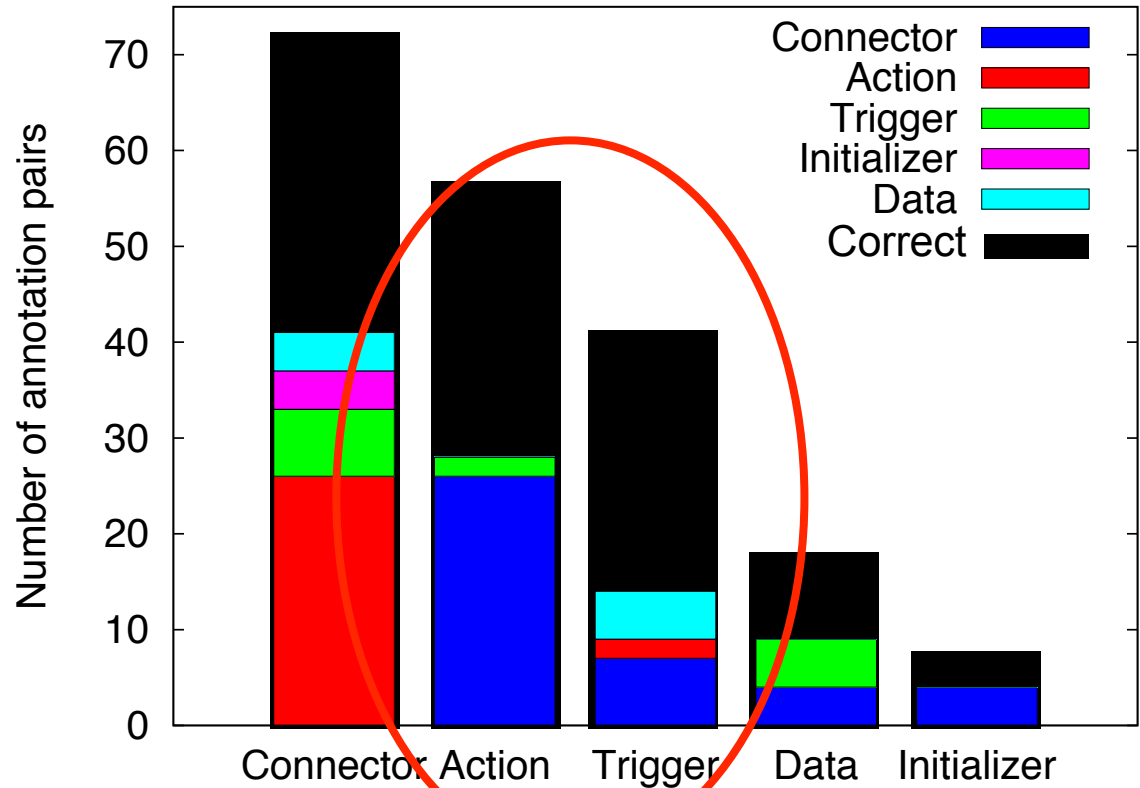
Evaluation

- 3 concerns from Jajuk, an open source iTunes
- Participants
 - **No roles:** 3 developers from prior study
[Robillard, Shepherd, Hill, Shanker, Pollock '07]
 - **Roles:** 5 developers from ABB + 1 UD Masters student
- Measures
 - % pairwise agreement (\cap / \cup)
 - Two sample t-test (when normal)
 - Mann-Whitney U-test (non-parametric)
 - Fleiss' kappa reliability measure (see paper)

RQ: How clear are the role definitions?

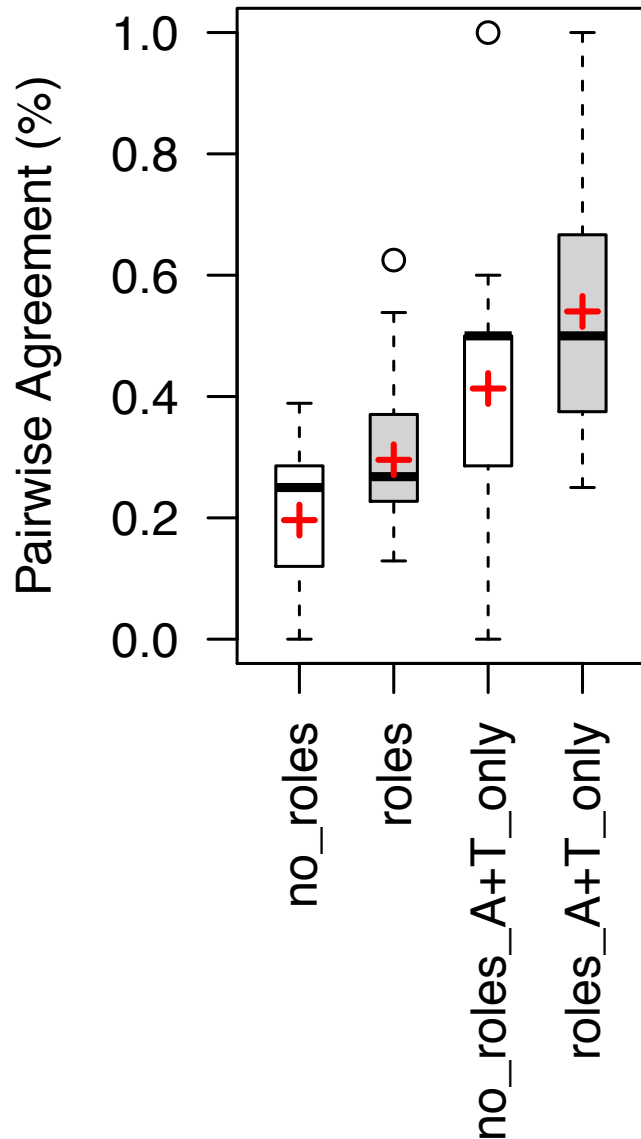
Conclusion:
Connectors are unclear & inconsistently annotated

Incorrect Annotation Pairs



Survey Question	μ	$\pm\sigma$
<i>How clear was the description of:</i>		
connector element roles?	4.67	1.03
action nodes?	5.67	1.03
trigger nodes?	5.83	0.98
initializer nodes?	5.33	1.21
data nodes?	5.00	1.67
connector nodes?	3.83	1.47

RQ: Do roles increase agreement?



Conclusions:

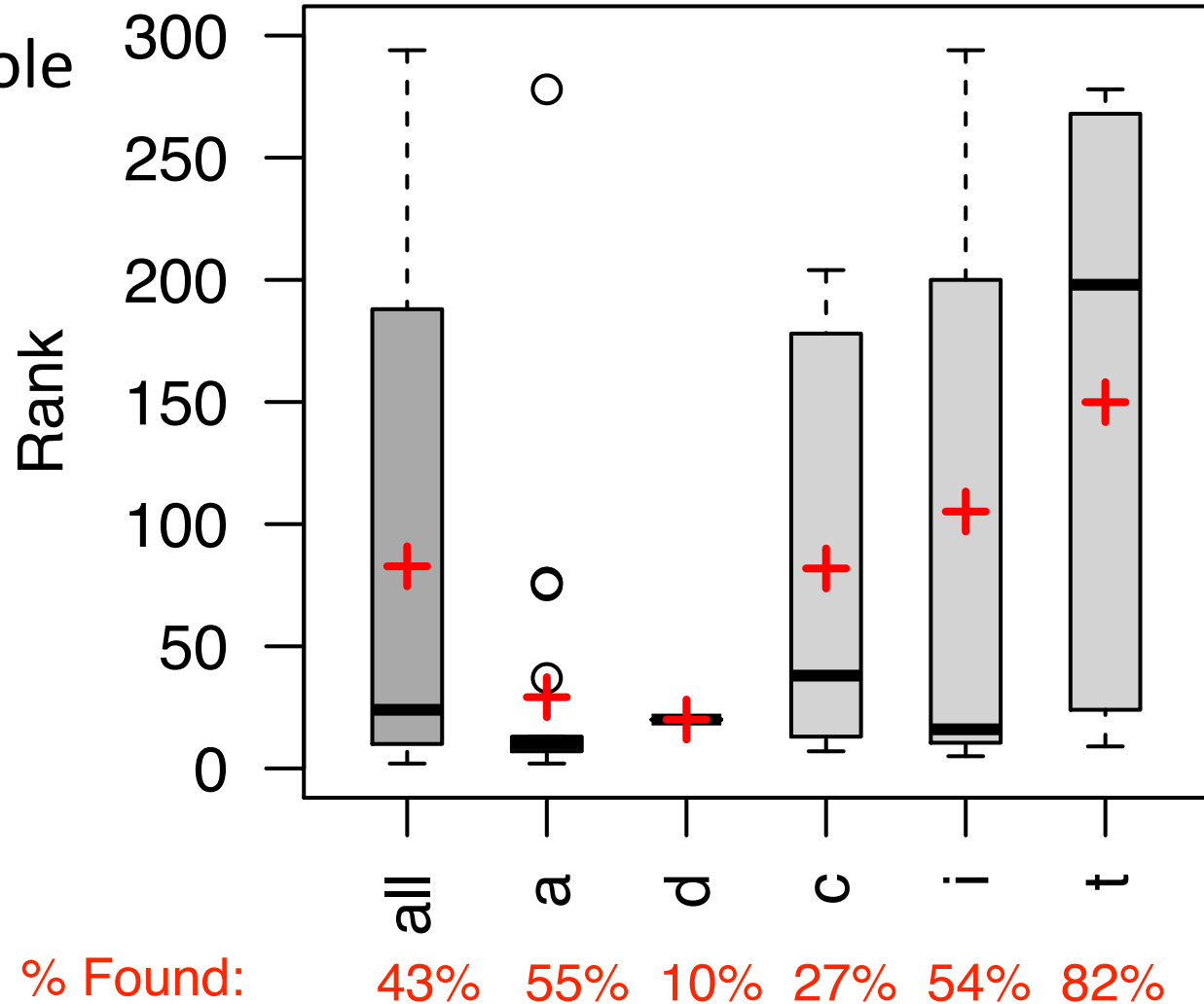
- Knowledge of roles has significantly higher agreement than no roles (t-test, $p < 0.05$)
- Action + trigger subset (roles A+T only) significantly outperforms roles with all elements (U-test, $p < 0.05$)

Exploratory Study: Insights into Feature Location Improvement

- **Tool:** DocFetcher (docfetcher.sourceforge.net)
Lucene-based tf-idf implementation
- **Gold set:** union of concern elements annotated by participants

RQ: What roles are found at rank...?

- Recall varies by role
- Actions have highest ranks
- Triggers most found, but worst ranks
- Future FL tool improvements:
 - Use call graph



Conclusion & **Community** Feedback

- **Gold set creation:** knowledge of roles leads to statistically significant increase in agreement
- **Feature location tool improvement:**
 - Recall varies widely by role
 - Roles provide additional semantic information
- **Discussion questions:**
 - How can we create better gold sets? Roles help...
 - Should FLT optimization take roles into account?

Exploratory Study: Insights into Feature Location Improvement

- **Tool:** DocFetcher (docfetcher.sourceforge.net)
Lucene-based tf-idf implementation
- **Gold set:** union of concern elements annotated by participants

Role	Number Found	Total Number	Percent Found	Recall @ 5	Recall @ 10	Recall @ 20
a	24	44	55%	0.11	0.30	0.43
d	1	10	10%	0.00	0.00	0.10
c	21	77	27%	0.00	0.05	0.10
i	7	13	54%	0.15	0.15	0.31
t	18	22	82%	0.00	0.05	0.05
Total	71	166	43%	0.04	0.12	0.20

Describing Concerns/Features

- Precise verb phrase (VP)

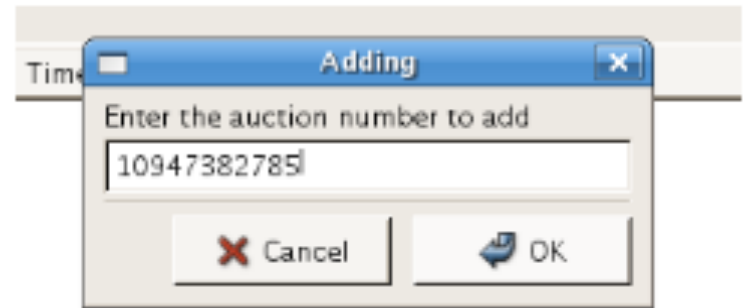
VP = verb + DO + IO*

e.g., **remove** **item** **from list**

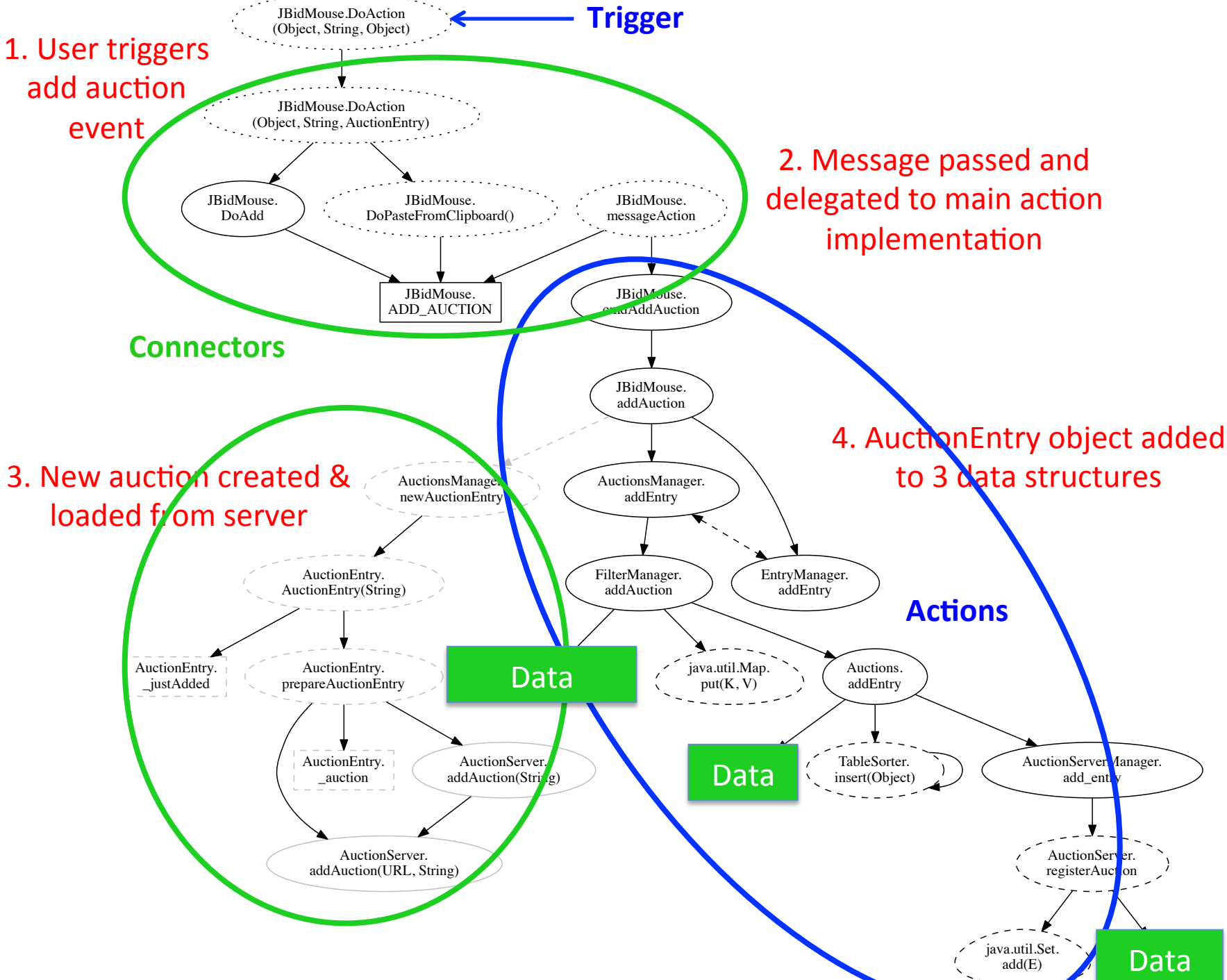
- DO: direct object
- IO: indirect object or modifiers (0 or more)
- Examples:
 - **✗** add auction ... *to what?*
 - **✓** add new auction to local system
 - shuffle songs by toggling playlist shuffle mode
 - sort music collection by genre

Example Concern:

Add an auction to jBidWatcher



Number	Current	Max	Time left	Description	Status	Seller
330006384386	\$1,030.55 (...)	--	3d, 7h	1980 RICKENBACKER 4003 BASS GUITAR FLA...		the_emperor_of_mars



Trigger Nodes

- Method or field that triggers action
 - Can trigger directly or through connectors
 - Doesn't implement concern's VP (those are actions)
 - Examples:
 - Field representing UI button
 - API method call from listener (e.g., actionPerformed)
 - Passing a message to event queue to be handled later
 - Delegating to a connector node

```
protected void DoAction(Object src, String actionString, AuctionEntry whichAuction,
    Component c_src;
    if(actionString.equals("Save")) DoSave(c_src);
    else if(actionString.equals("Load")) DoLoad(null);
    else if(actionString.equals("Add")) DoAdd(c_src);
    else if(actionString.equals("Delete")) DoDelete(c_src, whichAuction);
        MQFactory.getConcrete("thumbnail").enqueue(this);
    }
}
} catch(Exception e) {
    ErrorManagement.handleException("Error handling thumbnail loading", e);
}
```

Connectors

- Structurally connects and passes information between two concern nodes
- Often connect triggers to actions

1. Processes
the data (DP)

```
private void DoAdd(Component src) {  
    String endResult;  
    String prompt = "Enter the auction number to add";  
  
    endResult = promptString(src, prompt, "Adding");  
  
    // They closed the window or cancelled.  
    if (endResult == null) return;  
  
    endResult = endResult.trim();  
    MQFactory.getConcrete("user").enqueue(ADD_AUCTION + endResult);  
}
```

2. Passes data
& message

Action Nodes

- Directly implements concern's VP:
 - Using primitive types & operations
 - Call external libraries
 - Call more generic support methods within system
- Name may not exactly match VP

```
private /**
 * @brief Auctions must be registered when they are added, so that
 * the auction server can keep a list of auctions that it is
 * managing. This is used when storing out the list of auctions per
 * server.
 *
 * @param ae - The AuctionEntry to add to the server's list.
 */
public void registerAuction(AuctionEntry ae) {
    synchronized(_aucList) { _aucList.add(ae); }
}
```