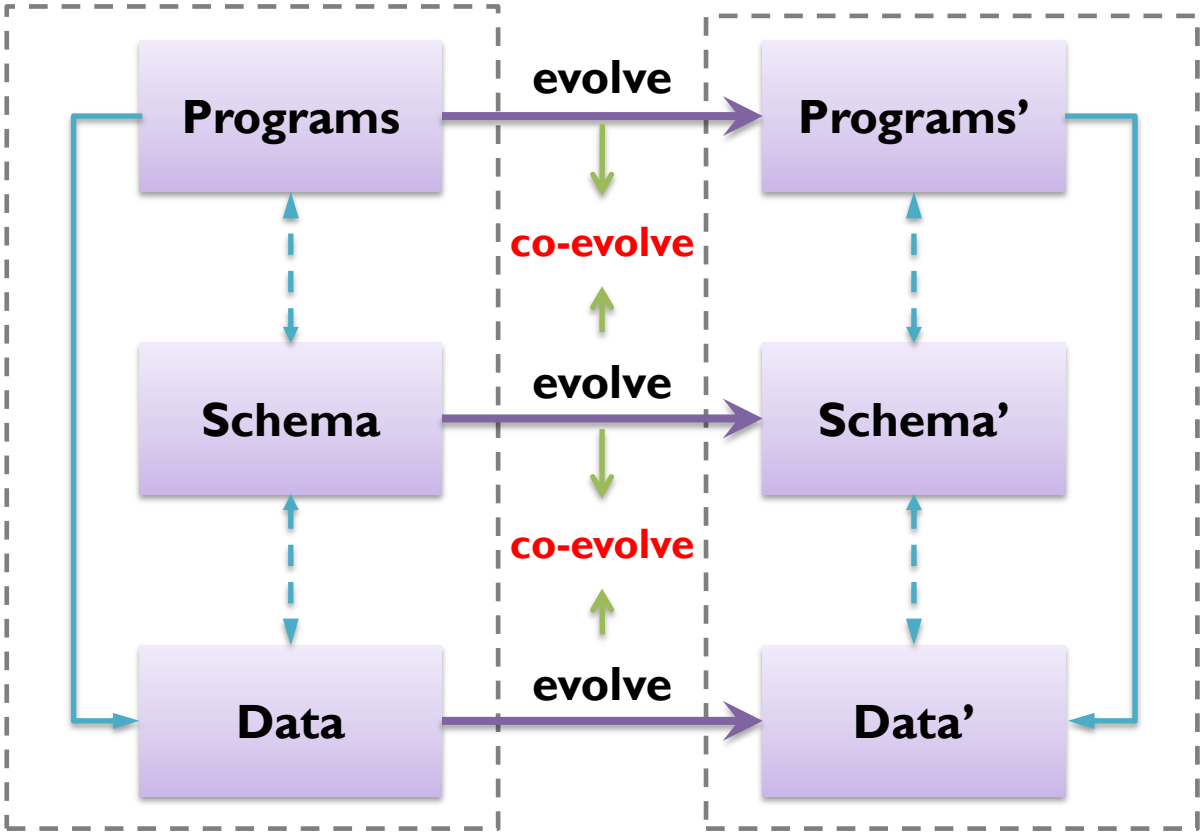
An Empirical Analysis of the Co-evolution of Schema and Code in Database Application

Dong Qiu*, Bixin Li*, Zhendong Su#

* Southeast University, China
# University of California, Davis, USA

Database Applications

Co-evolution



Key Contributions

- **First** large, comprehensive study of how schema & code co-evolve

- Guidelines to help automate the co-evolution process

Result Highlights

- Database schemas evolve **frequently** during the application lifecycle
- Schema changes induce **significant** code-level modifications
- Co-change analyses can be viable to automate or assist with database application evolution

Study Subjects

Project	History (Year)	Revisions (K)	#Tables	LoC (K)
coppermine	8.0	8	8~22	27~300
dotproject	7.3	5	15~63	8~150
e107	9.4	12	33~30	36~150
joomla!	6.4	23	35~61	10~250
mediawiki	3.1	42	3~51	3~880
prestashop	10	14	113~157	11~230
roundcube	4.5	6	5~12	20~120
tikiwiki	6.8	40	20~242	10~1,240
typo3	5.6	10	10~18	78~440
webERP	8.3	5	63~122	36~210
total / average	**69.4 / 6.9**	**160 / 16**		

Research Questions

RQ1: How frequently and extensively do database schemas evolve?

RQ2: How do database schemas evolve?

RQ3: How much application code has co-changed with a schema change?

Analysis Process

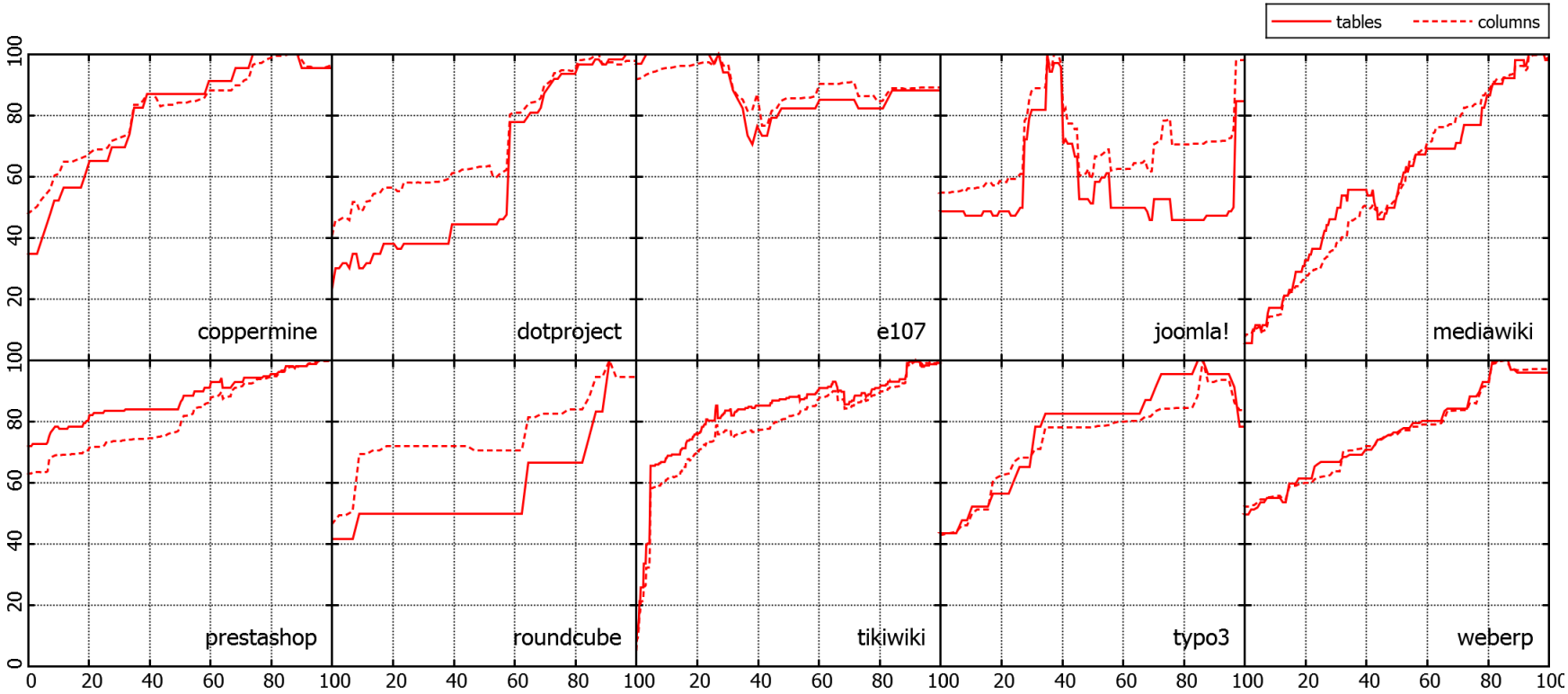
- Locate schema files
- Extract DB revisions
- Extract valid DB revisions
- Extract schema changes
- Co-change Analysis

RQ1 Results

Schemas evolve frequently

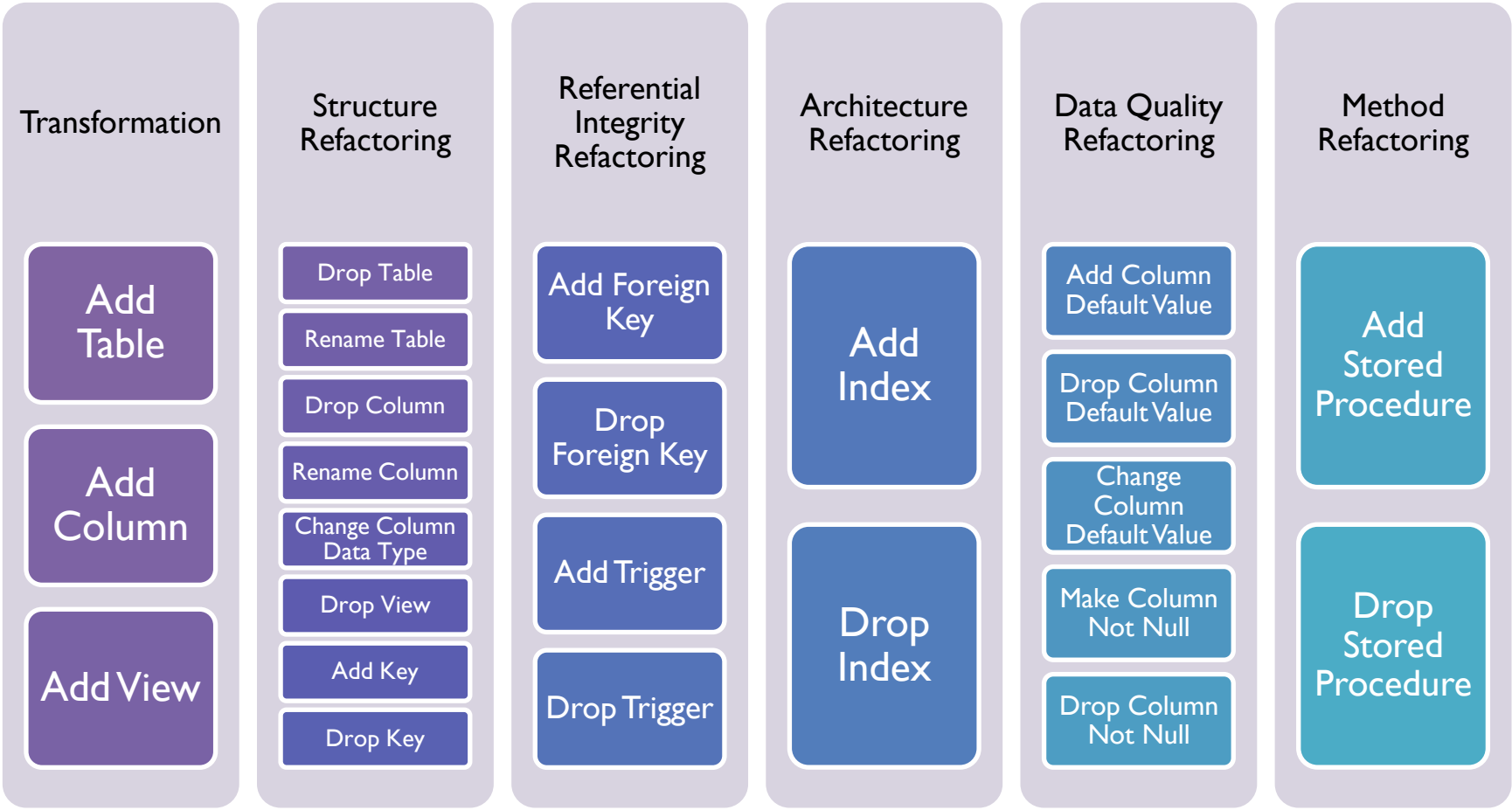
- Per release
  - On average **15** valid DB revisions
  - On average **67** atomic schema changes
- Per year
  - On average **21** valid DB revisions
  - On average **90** atomic schema changes

RQ1 Results

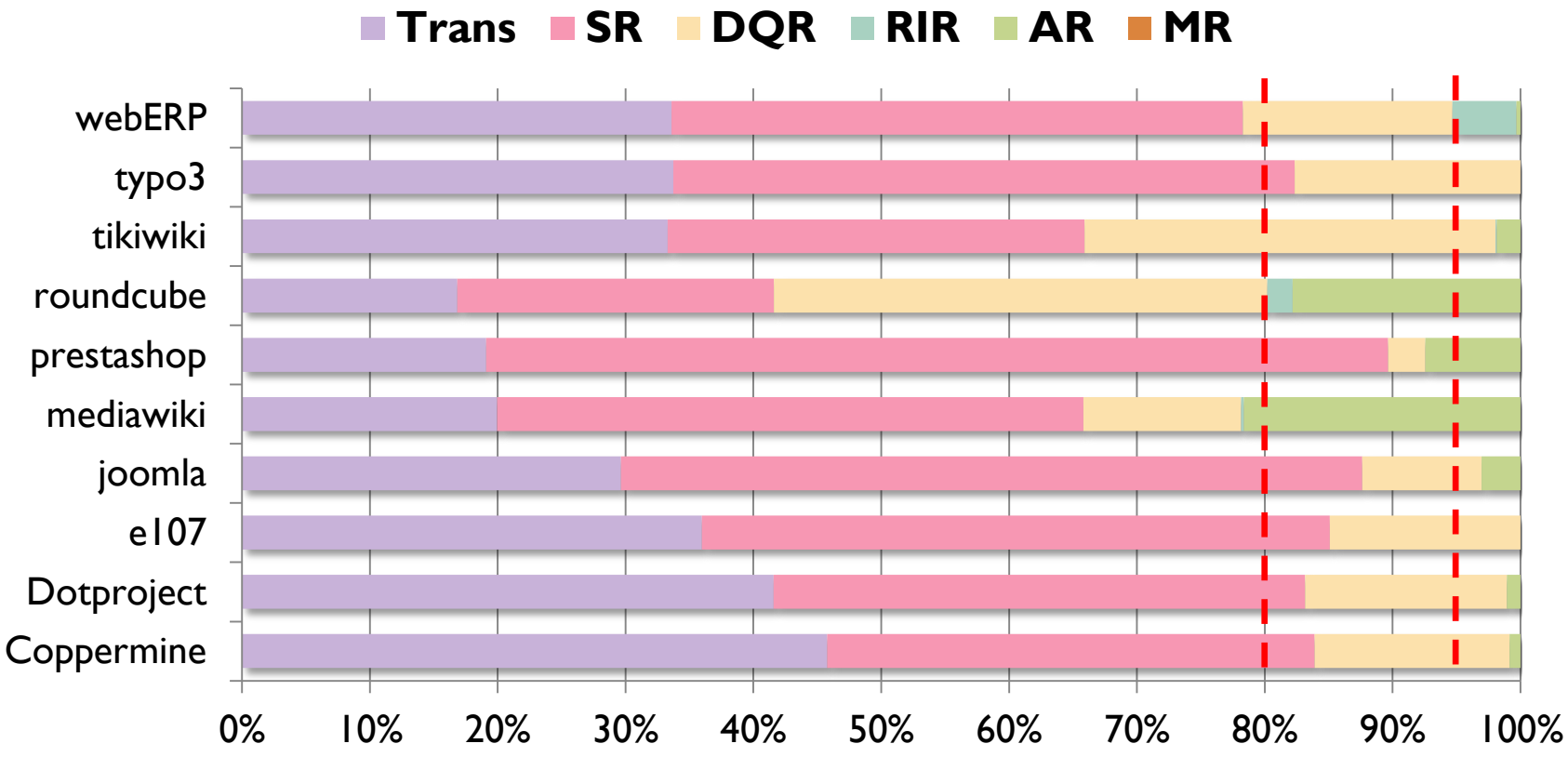


- Most projects' schemas grew significantly in size
- Two Projects' schemas show frequent fluctuations
- Tables and columns show similar change trend

Schema Change Categories

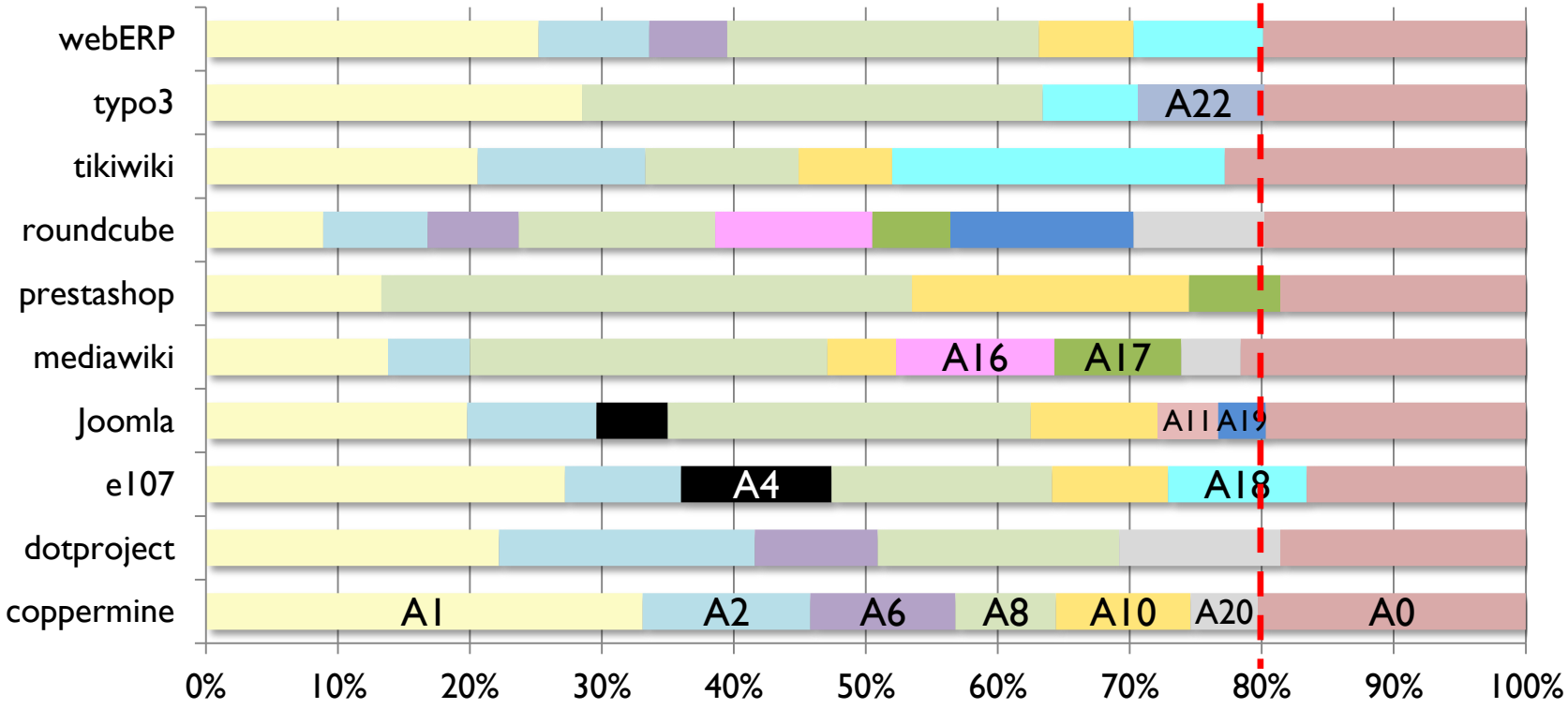


Distribution on High-Level Schema Changes



- **Trans**, **SR** and **DQR** covered most of the schema changes
- **AR** occurred relatively infrequently in some projects

Distribution on Low-Level Schema Changes

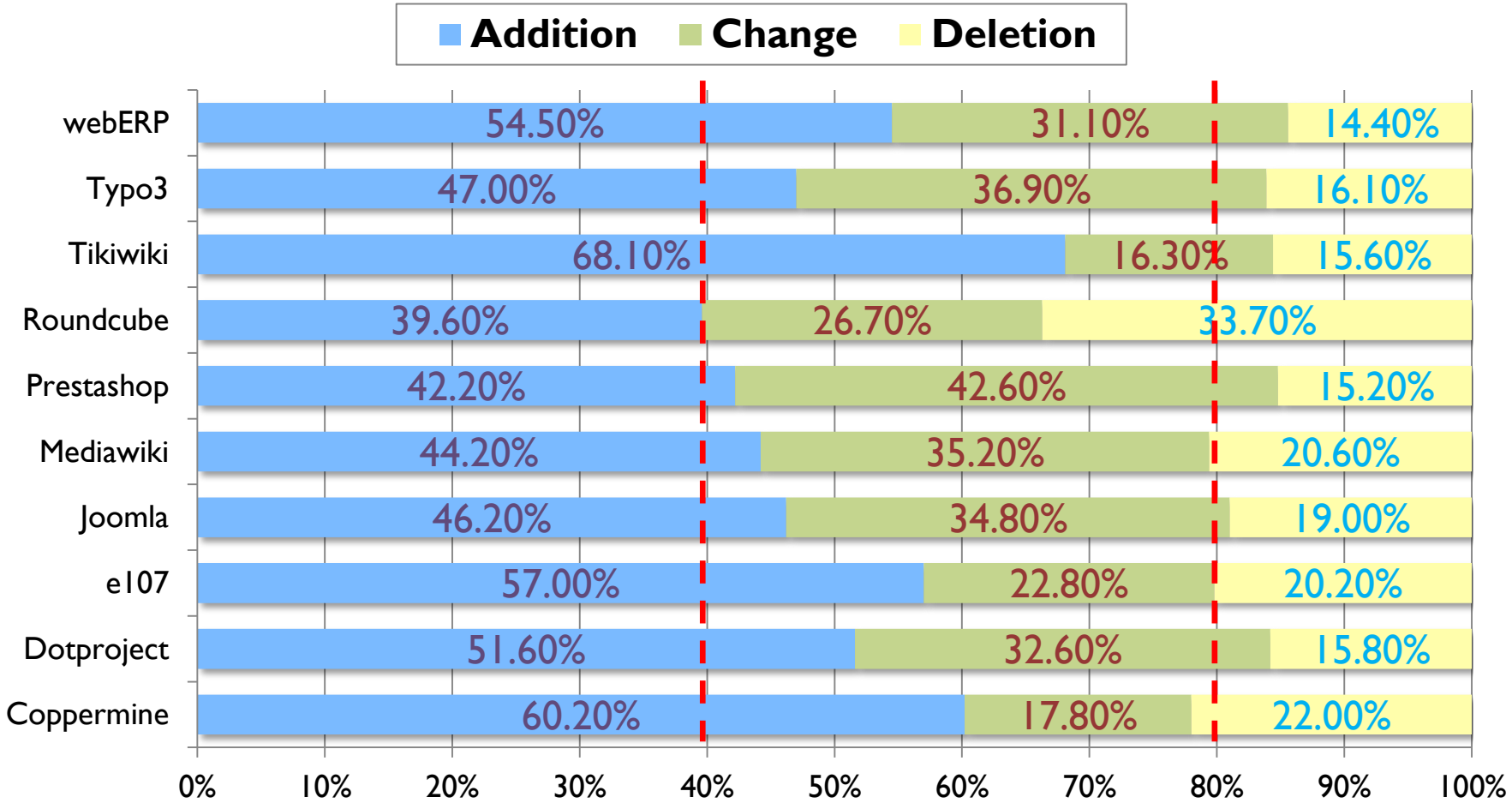


ID	Change Type	ID	Change Type	ID	Change Type	ID	Change Type
A1	Add Table	**A8**	Change Column Data Type	**A16**	Add Index	**A19**	Drop Column Default Value
A2	Add Column			**A17**	Drop Index	**A20**	Change Column Default Value
A4	Drop Table	**A10**	Add Key	**A18**	Add Column Default Value	**A22**	Drop Column Not Null
A6	Drop Column	**A11**	Drop Key			**A0**	Others

RQ2 Results

- Rarely appeared changes
  - Foreign key related
- Never occurred changes
  - View related
  - Trigger related
  - Procedure related

Distribution on Addition/Deletion/Change



Addition and **Change** accounted for most of the schema evolution

Co-change Analysis

- Customers who bought this item **also** bought these other items
- Programmers who changed this source code **also** changed this other code
- Programmers who changed database schemas **also** changed other application code

Co-change Analysis Hypothesis

In a valid DB revision

all program code-level changes

from the **same revision** are considered

driven by schema changes

Co-change Analysis Validity

- Performed a detailed manual analysis on **10%** of randomly selected valid DB revisions

- Verified that
  - **72%** valid DB revisions provide useful co-change info
  - **>80%** valid DB revisions have precision **>60%**
  - **~56%** valid DB revisions have **100%** precision

$$precision(r) = \frac{|RC_r \cap CC_r|}{CC_r}$$

RQ3 Results

- For each atomic change, **10-100** LoC were changed on average

- For each valid DB revision, **100~1000** LoC were changed on average

Estimated Changed Code Size

Project	$\hat{\beta}_{Tran}$	$\hat{\beta}_{SR}$	$\hat{\beta}_{RIR}$	$\hat{\beta}_{AR}$	$\hat{\beta}_{DQR}$	$\hat{\beta}_{MR}$	$\overline{R^2}$
Coppermine	157	10	0	0	18	NA	0.50
Dotproject	105	27	0	1	0	NA	0.80
E107	35	13	0	0	17	NA	0.32
Joomla!	243	54	0	0	41	NA	0.64
Mediawiki	112	16	0	4	0	NA	0.38
Prestashop	103	29	0	0	17	NA	0.34
Roundcube	323	45	0	35	0	NA	0.64
Tikiwiki	232	36	0	50	13	NA	0.51
TYPO3	85	200	0	0	0	NA	0.32
webERP	35	33	0	8	0	NA	0.32

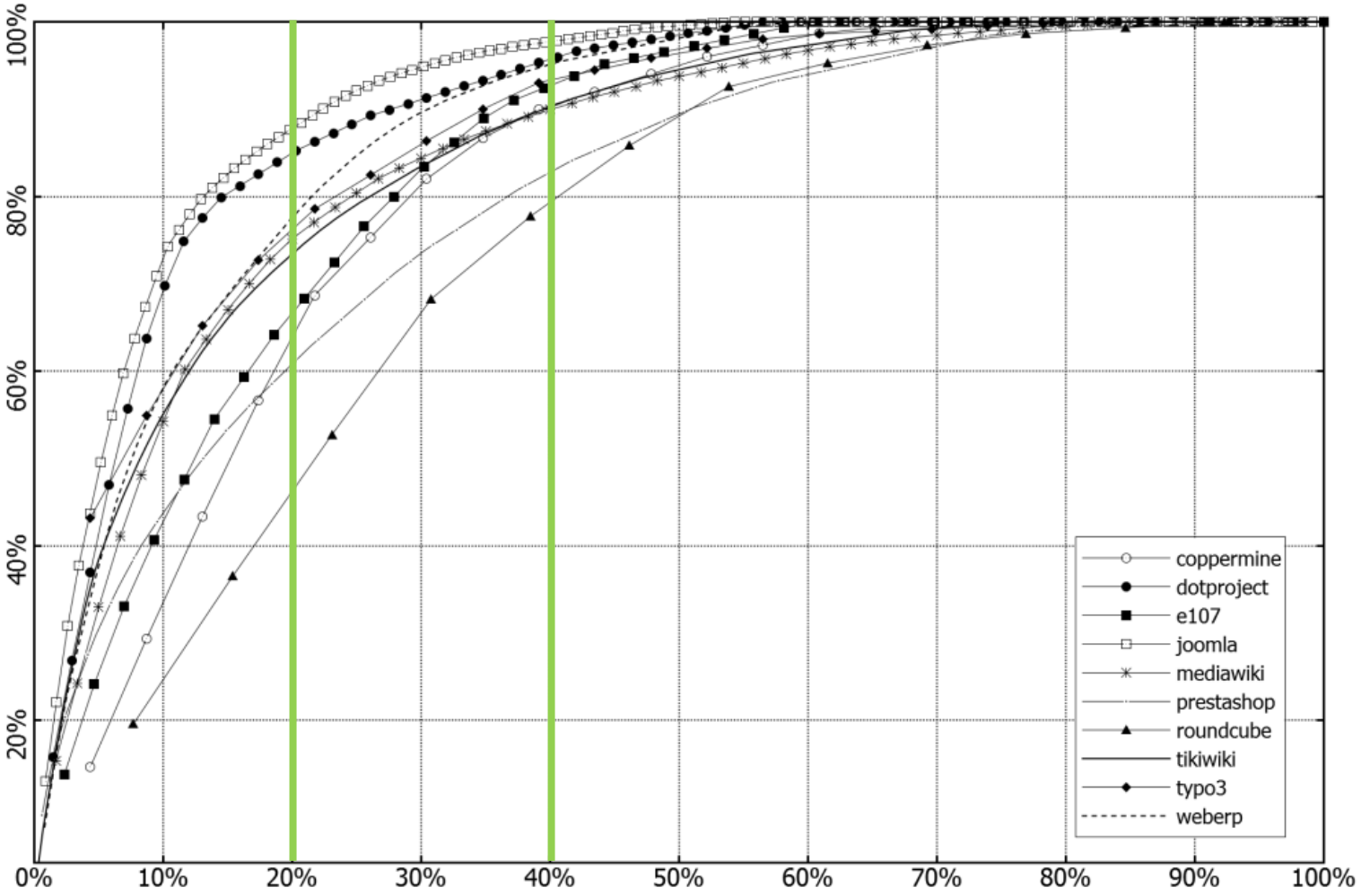
$$y = \sum_i \beta_i x_i$$

Tool Support

- Desirable features
  - Before a schema change, find impacted code region
  - For a schema change, locate all impacted code region
  - Guide developers to evolve code

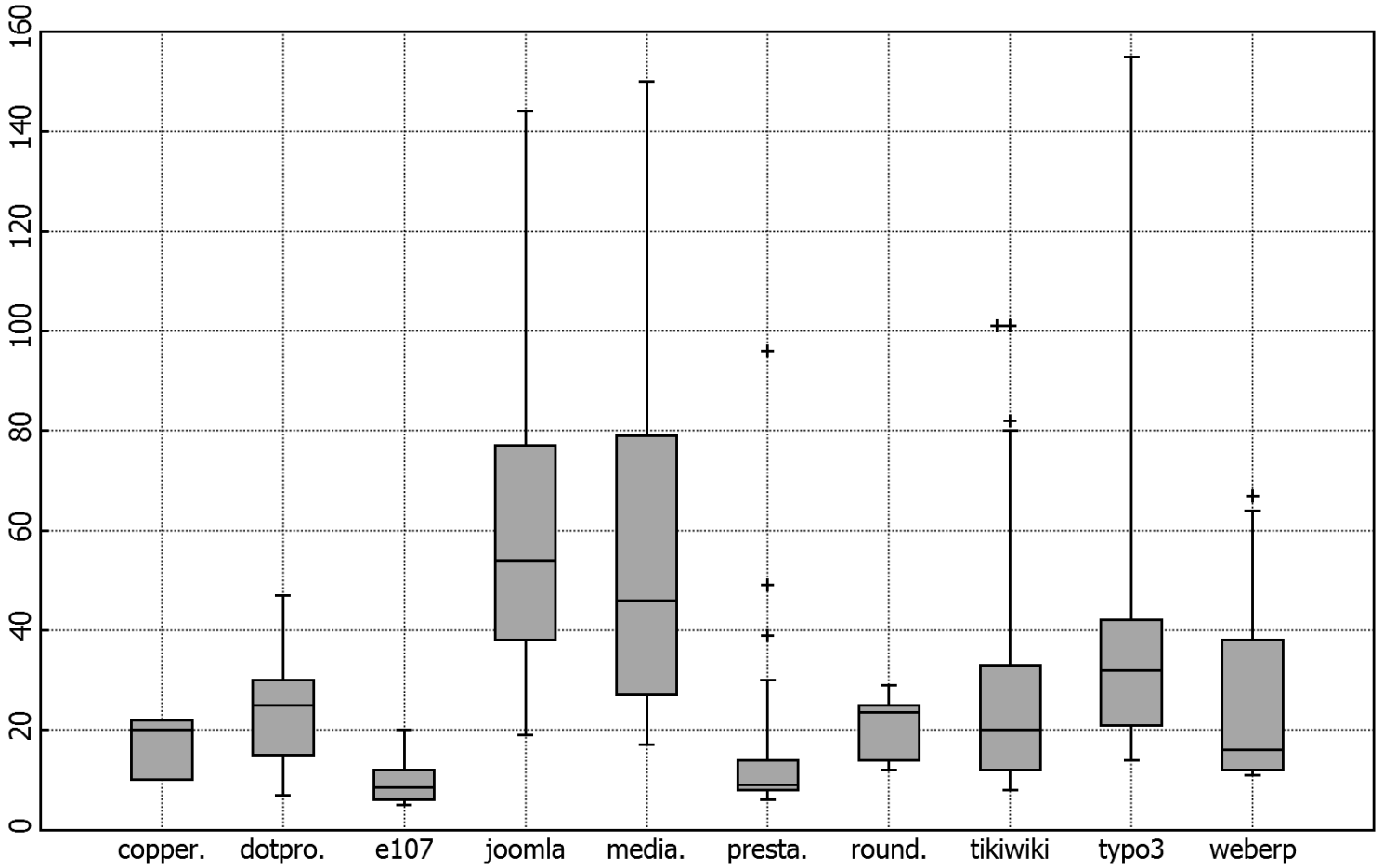
- Promising approaches for **automation**
  - Change impact analysis
  - Co-change analysis

Coverage of Schema Changes by Tables



Schema changes are **centralized**

Atomic Schema Changes Per Table



The histories of frequently changed tables are sufficient.

Conclusion

- Presented the first large-scale empirical analysis of how schema and code co-evolve

- Exposed new, useful quantitative knowledge to help automate the co-evolution task

Thank you!

Data: http://ise.seu.edu.cn/people/dongqiu/