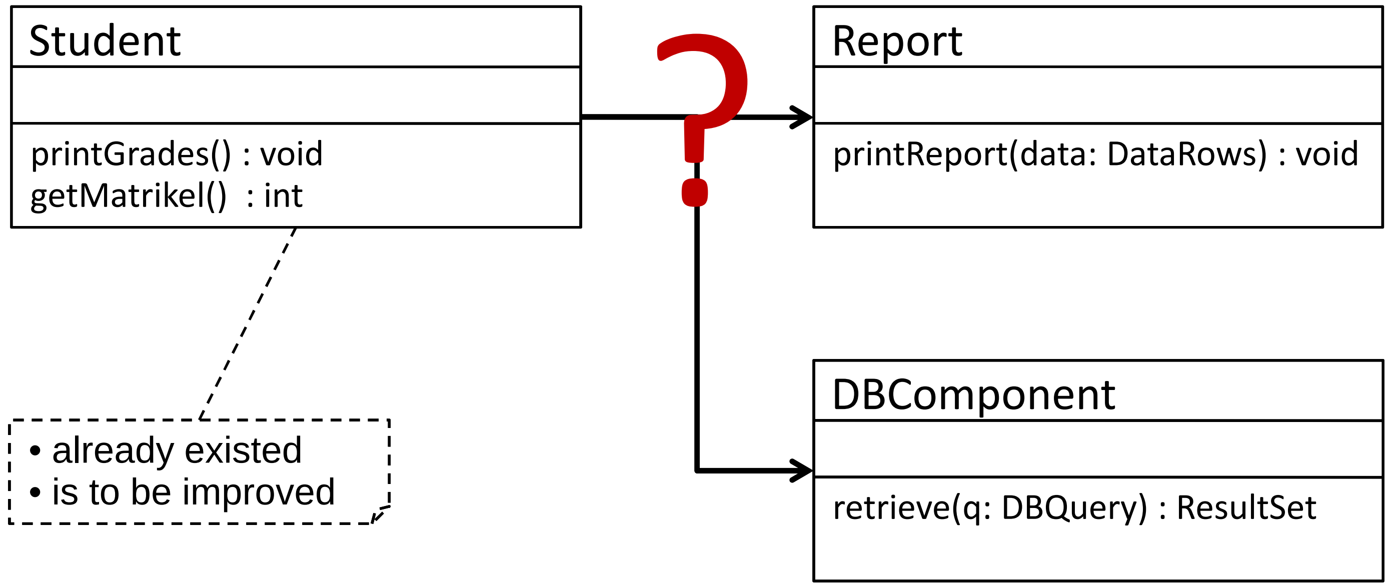
TECHNISCHE UNIVERSITÄT DRESDEN

Exploring Role-Based Adaptation

Sebastian Götz, Defense “Großer Beleg” 20.11.08

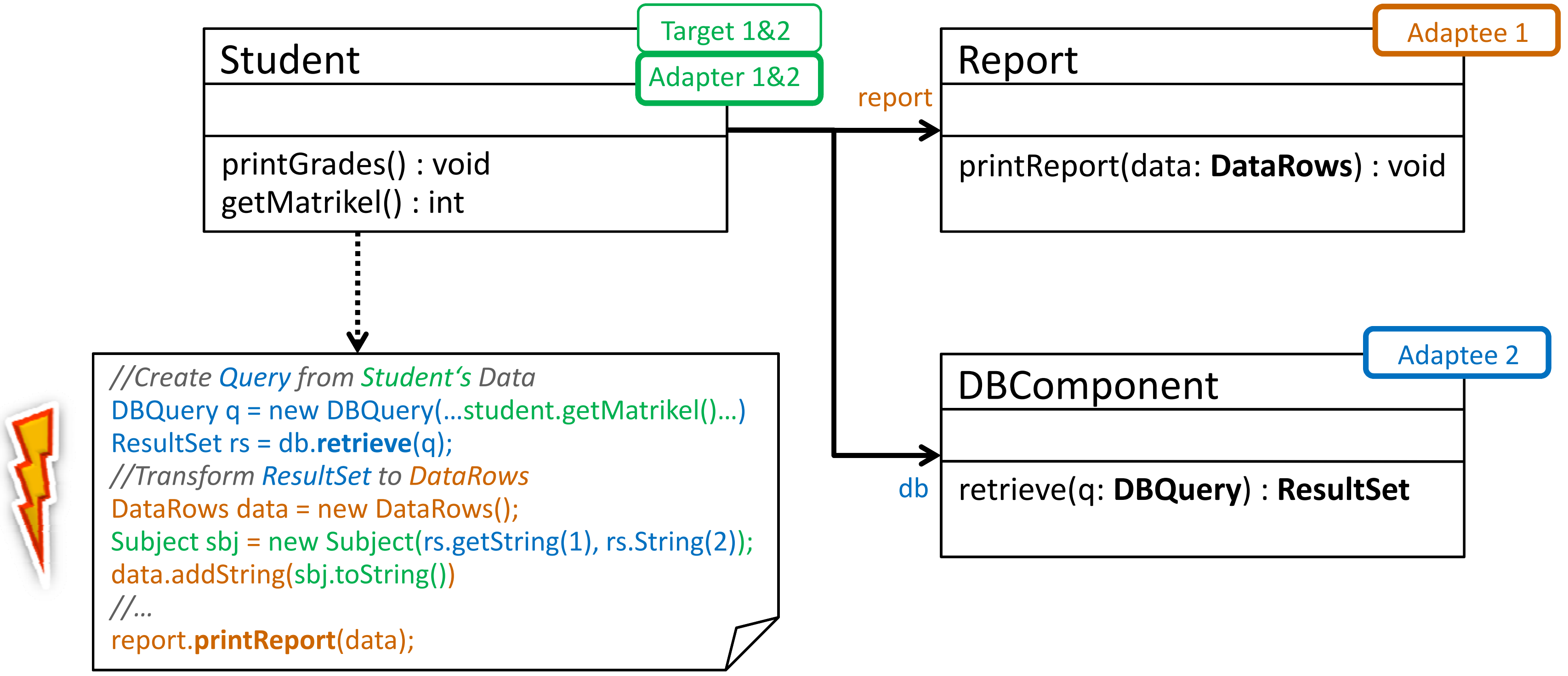
Outline

1. Motivation



1. Motivation

➔ Adapter Design Pattern [1] applied twice
➔ **intertwined code**

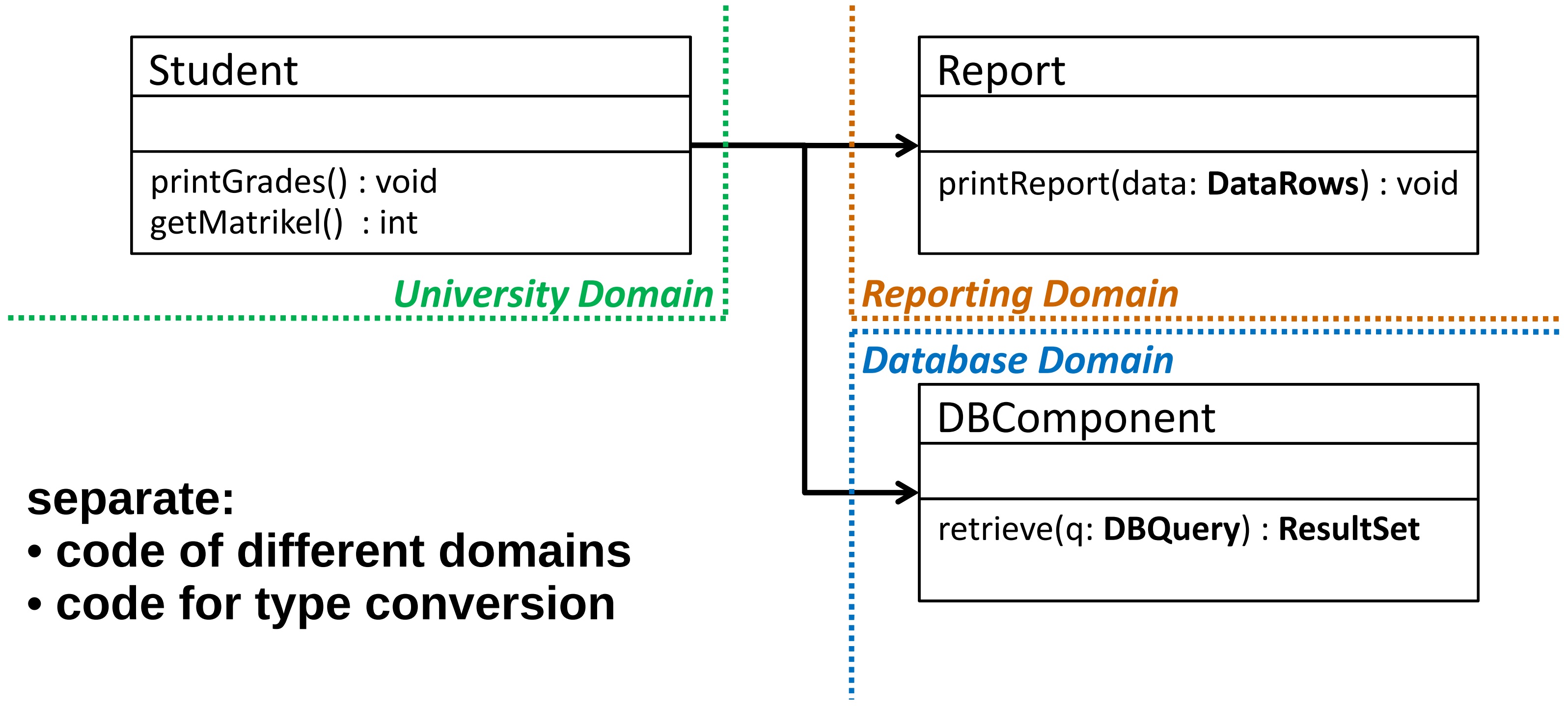


```
//Create Query from Student's Data
DBQuery q = new DBQuery(…student.getMatrikel()…)
ResultSet rs = db.retrieve(q);
//Transform ResultSet to DataRows
DataRows data = new DataRows();
Subject sbj = new Subject(rs.getString(1), rs.String(2));
data.addString(sbj.toString())
//…
report.printReport(data);
```

Roles annotated as in [2]

1. Motivation

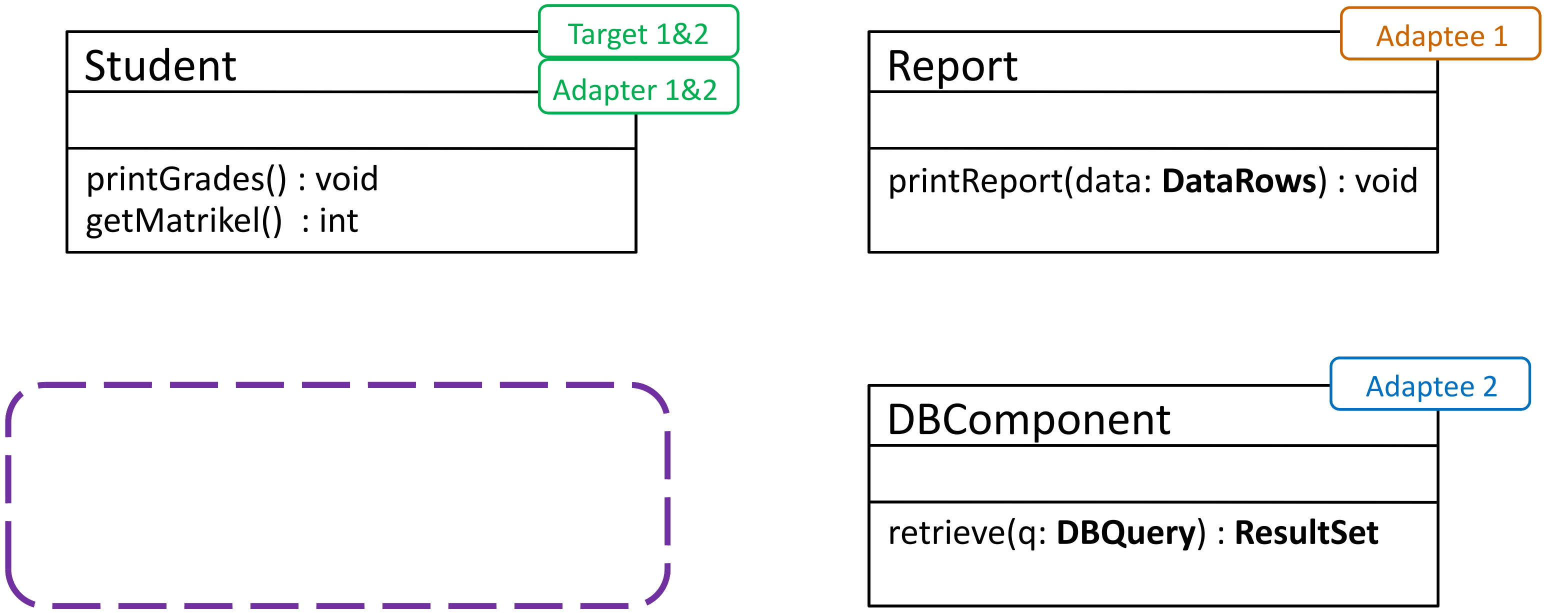
Goal: separation of adaptation concerns



separate:
- **code of different domains**
- **code for type conversion**

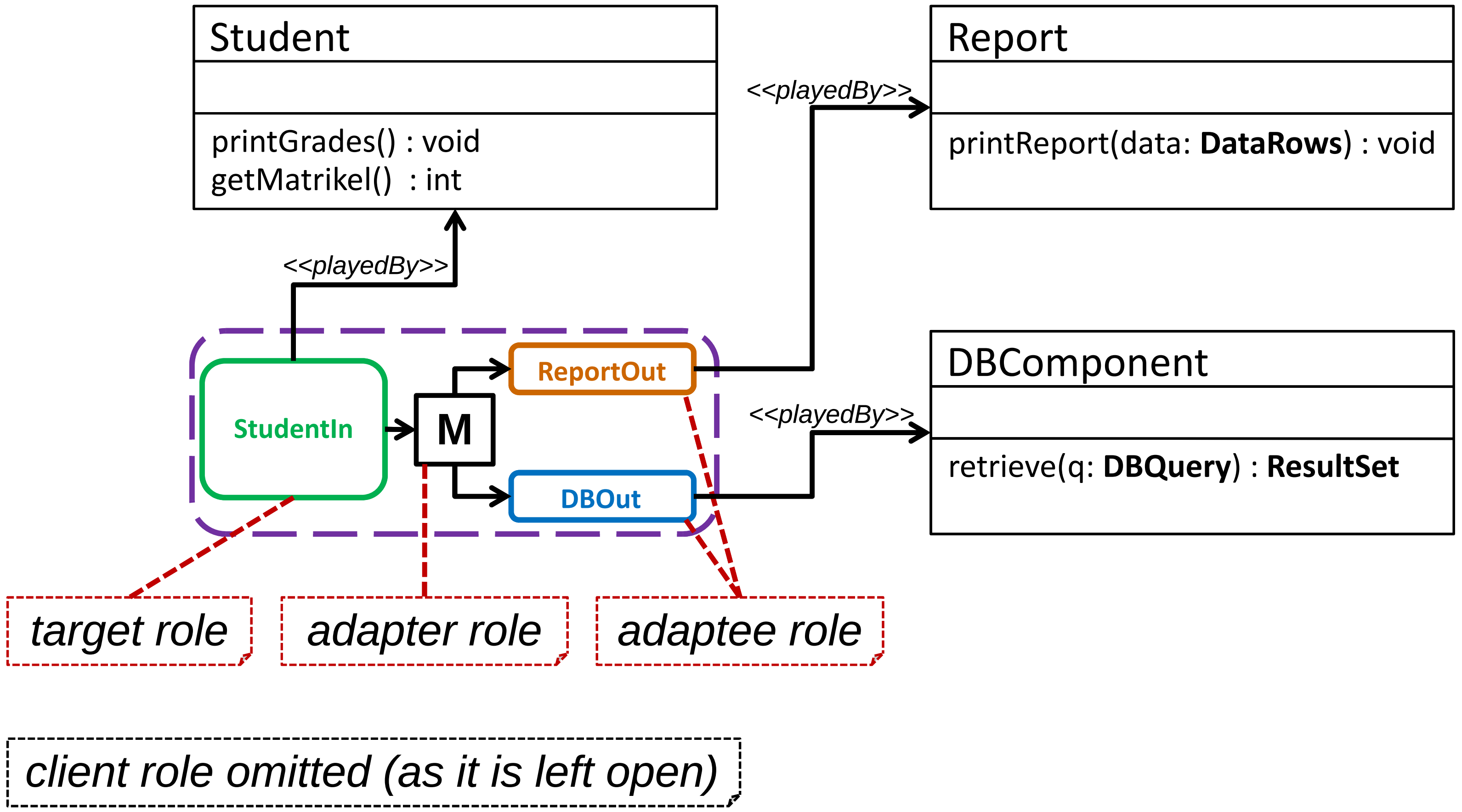
2. Elements of Role-based Adapters

➔ implement adapter roles as first-order programming constructs



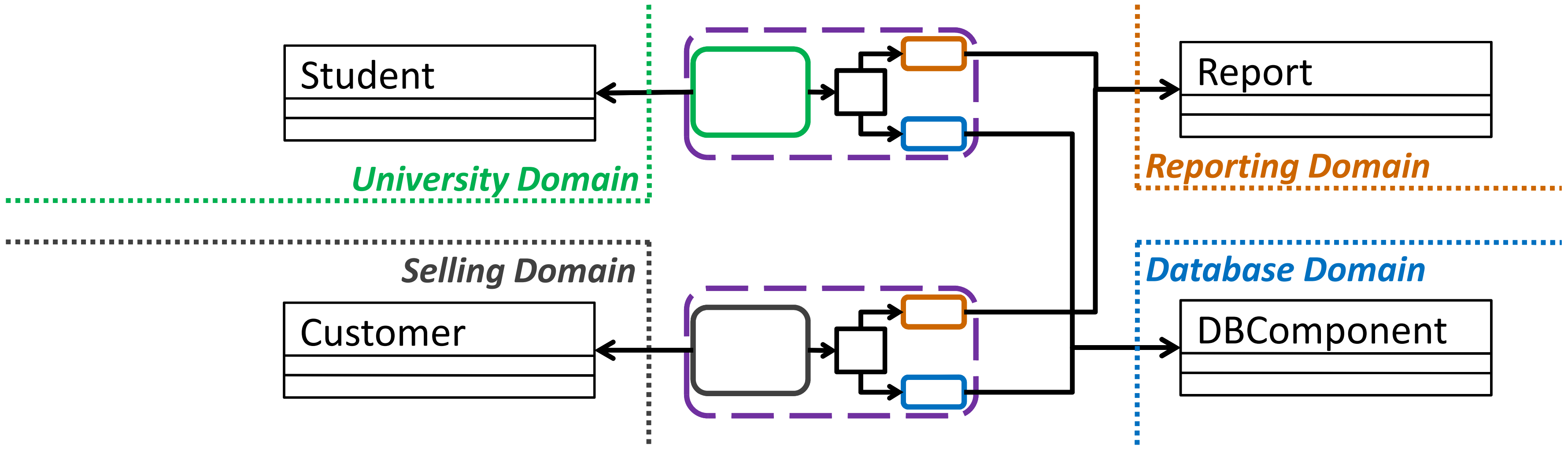
2. Elements of Role-based Adapters

➔ split adapter



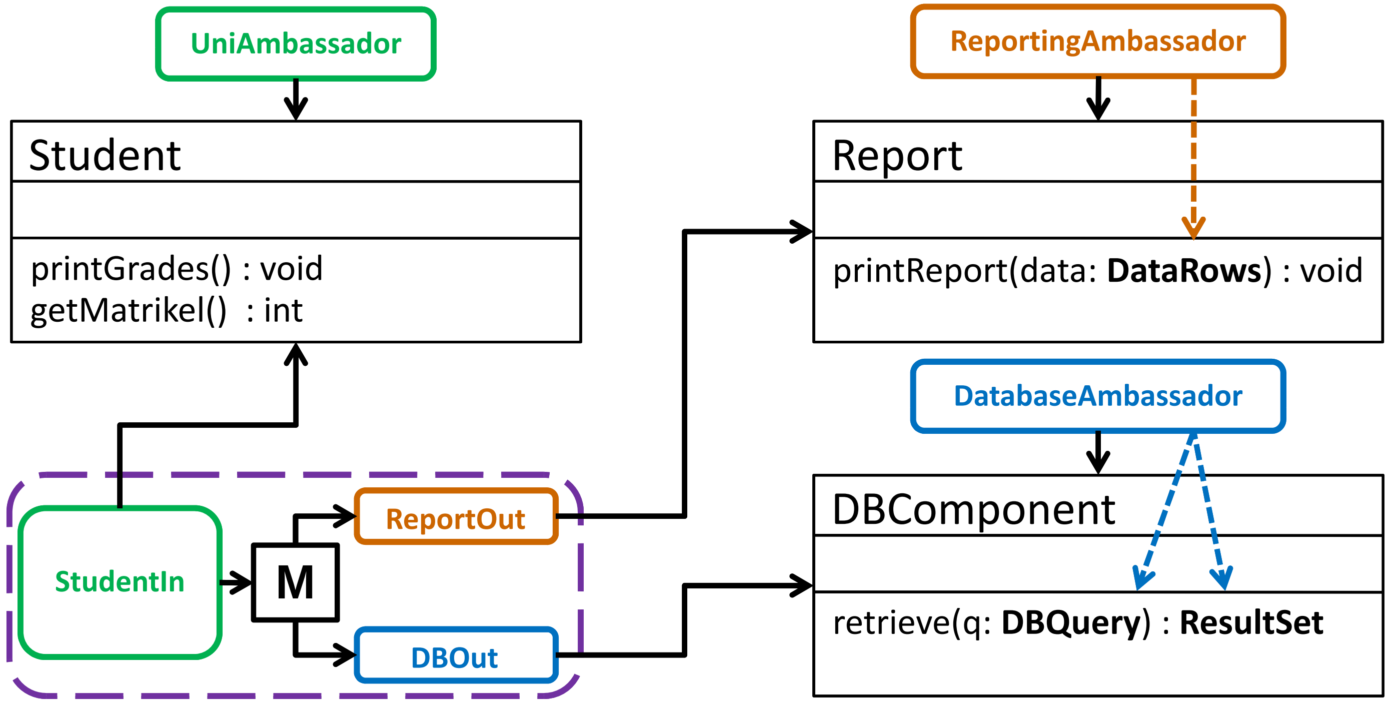
2. Elements of Role-based Adapters

➔ type conversion implemented twice (for Report and DB component)
➔ code replication



➔ type conversion should not be the task of In-/Out-Roles

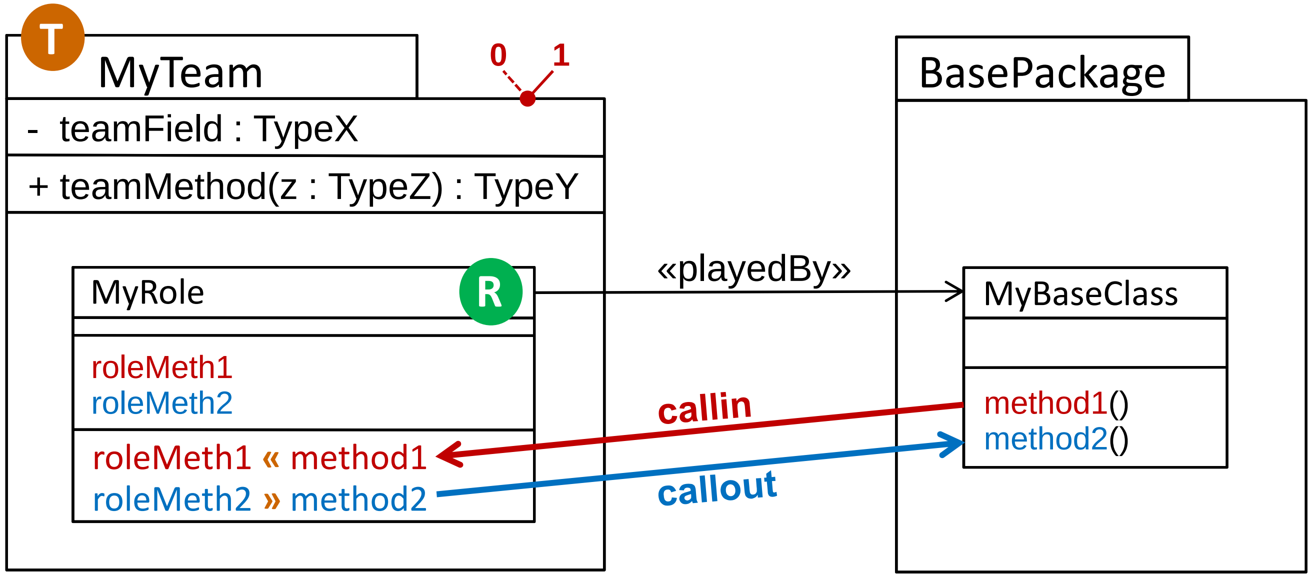
2. Elements of Role-based Adapters



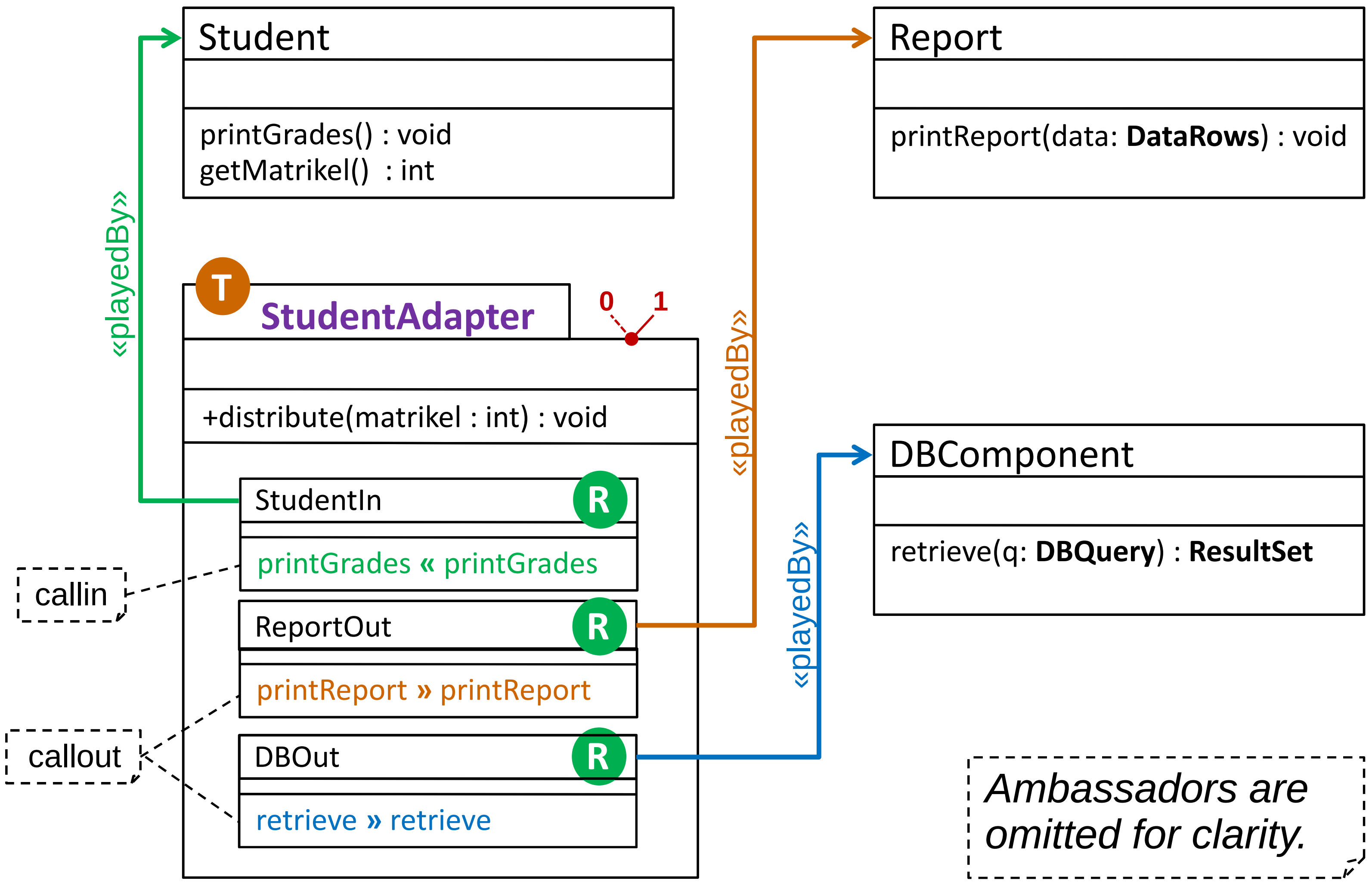
➔ Ambassadors for type conversion
➔ no code replication
➔ reuseable

3. Realization in ObjectTeams/Java

➔ use ObjectTeams [3]
➔ **roles** are played by **actors** in a **context** [4]



3. Realization in ObjectTeams/Java



Student
printGrades() : void
getMatrikel() : int
Report
printReport(data: **DataRows**) : void
T
StudentAdapter
0
1
+distribute(matrikel : int) : void
«playedBy»
«playedBy»
«playedBy»
StudentIn
R
printGrades « printGrades
ReportOut
R
printReport » printReport
DBOut
R
retrieve » retrieve
DBComponent
retrieve(q: **DBQuery**) : **ResultSet**
callin
callout
Ambassadors are omitted for clarity.

Demonstration

5. Conclusion

Advantages	Disadvantages
• Less effort for maintenance	• More initial coding effort
• More flexible allocation of developers	• More initial training effort
• Easier development of adapters	
• (Reusable Ambassadors)	

Published as ECOOP-workshop paper [5] at RAM-SE 2008

6. Future Work

- **Collection of Empirical Data Using an Experiment**
- **Parameterizable RBAs**
- **Using RBAs for Change Encapsulation**

Thank you very much for your attention.

Any Questions ?

5. Future Work

- **Collection of Empirical Data Using an Experiment**

  - 2 Teams of 5 Students each

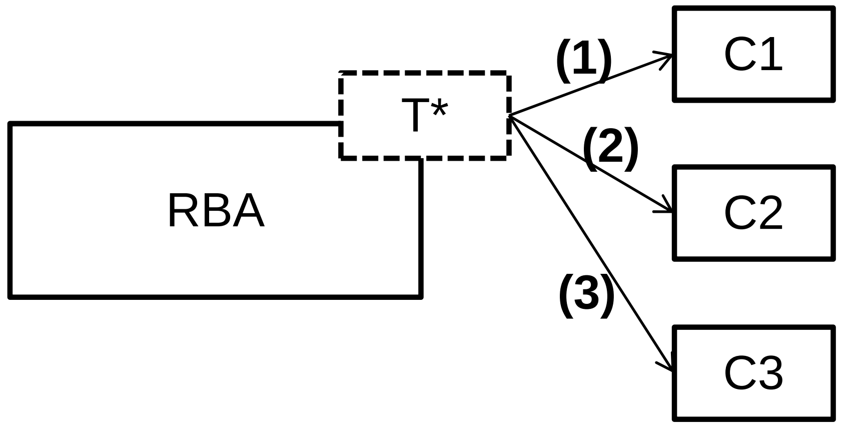
  1st Phase:
  - 1st Team develops the UMS using class-based adapters
  - 2nd Team uses RBAs

  2nd Phase:
  - Both Teams incorporate a set of changes

  Time measurement, Software Metrics and Surveys after 1st and 2nd Phase.

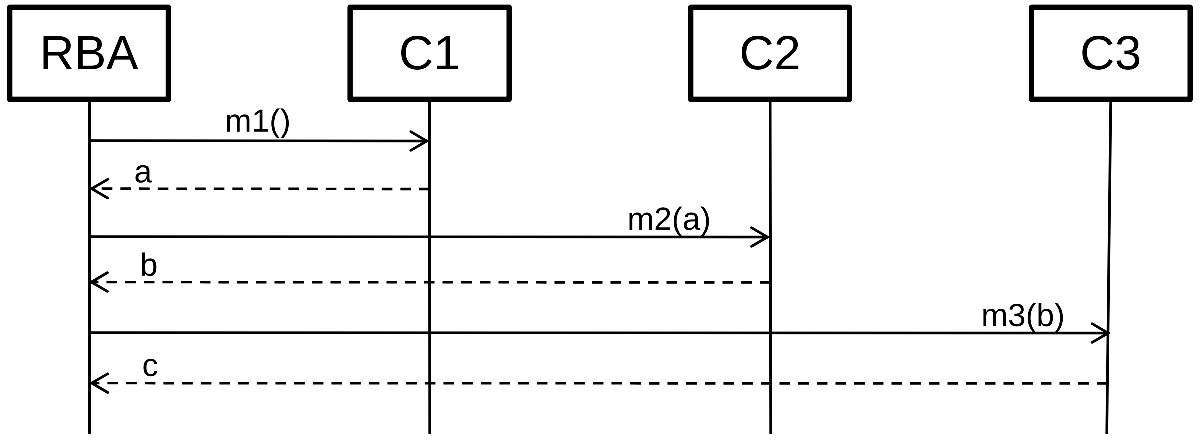
5. Future Work

- **Parameterizable Role-based Adapters**
  - RBAs take an ***ordered*** set of components as parameter



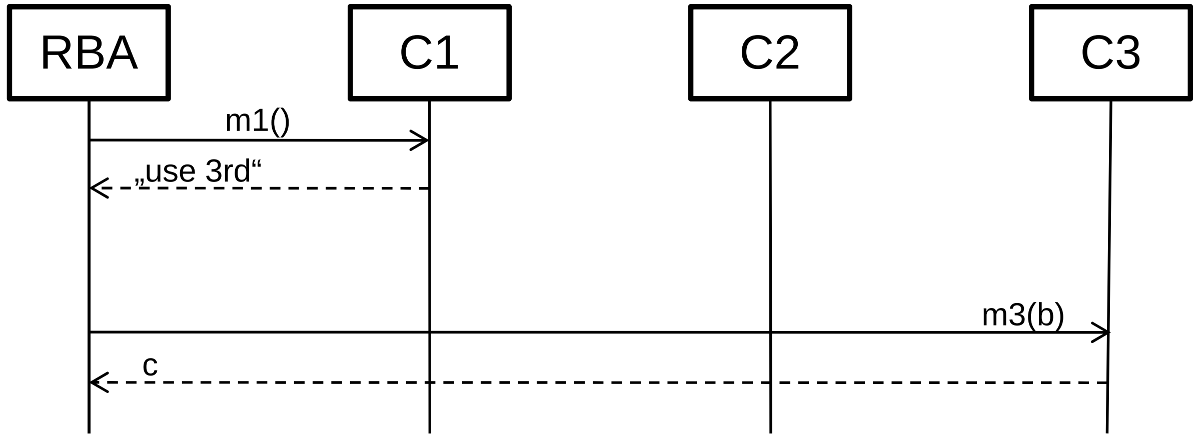
5. Future Work

- **Sequence RBA**



Return value of C(n) is passed as argument to C(n+1)

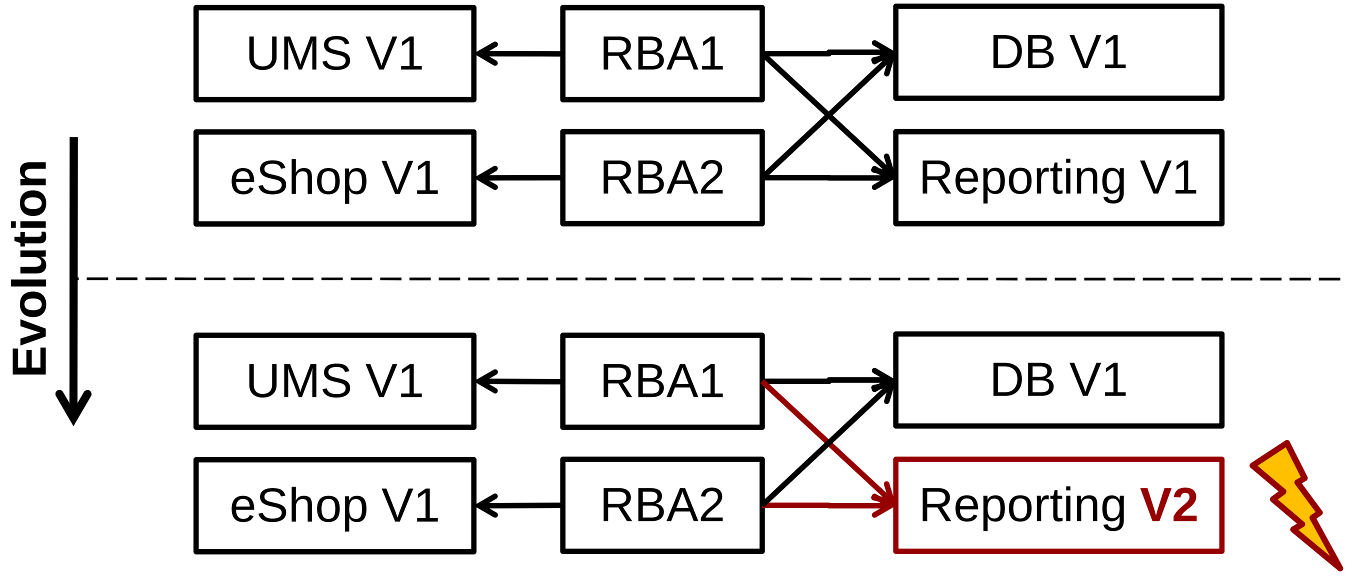
- **Pick RBA**



C(1) decides, which C(n) to call

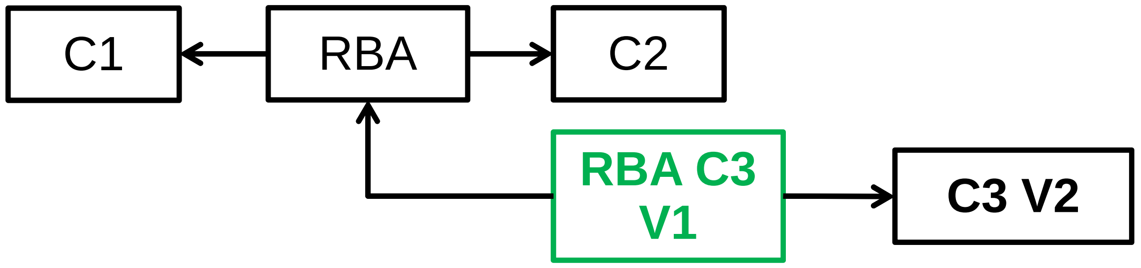
5. Future Work

- **Using Role-based Adapters for Change Encapsulation**



  - Easy location of positions in code to change
  - But **multipe** RBAs need to be adjusted

• Using Role-based Adapters for Change Encapsulation



- Put an RBA in front of the new version
- Adapt new version to old version

➔ Role-based realization of **ComeBack**-adapters [6]

Literature

[1] Gamma, E., Helm, R., Johnson, R., Vlissides, J.: **Design Patterns: Elements of Reusable Object-Oriented Software.** Addison-Wesley, Reading, Massachusetts (1995)

[2] Riehle, D.: **Framework Design – A Role Modeling Approach.** PhD-Thesis, Swiss federal institute of technology, Zurich. 2000.

[3] Herrmann, S., Hundt, C., Mosconi, M.: **ObjectTeams/Java Language Definition - version 1.0.** Technical Report 2007/03, Technical University Berlin (2007)

[4] Steimann, F.: **Formale Modellierung mit Rollen.** Habilitationsschrift, Universität Hannover. 2000

[5] Götz, S., Savga, I.: **Exploring Role-Based Adaptation**. In Proceedings of the 5th ECOOP'2008 Workshop on Reflection, AOP and Meta-Data for Software Evolution, RAM-SE '08. 2008

[6] Savga, I., Rudolf, M., Götz, S., Aßmann, U.: **Practical Refactoring-based Framework Upgrade.** In Proceedings of the 7th Internaional Conference on Generative Programming and Component Engineering, GPCE'08, 2008.