Reflection

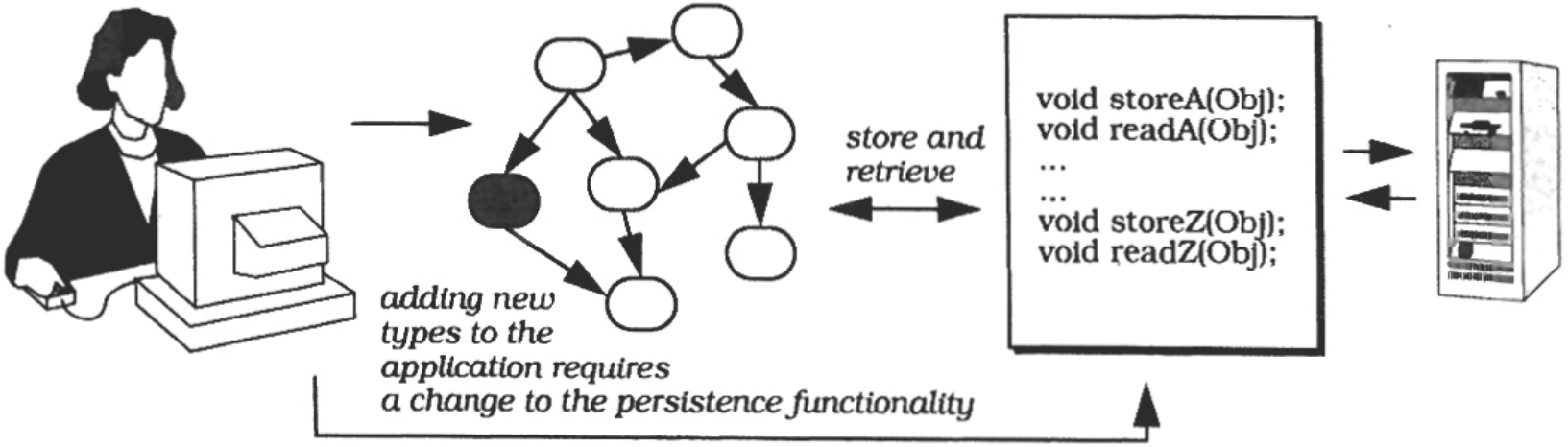
also known as : Open Implementation

The Reflection architectural pattern provides a mechanism for changing structure and behaviour of software systems dynamically.

Building systems that support their own modification a priori

Problem

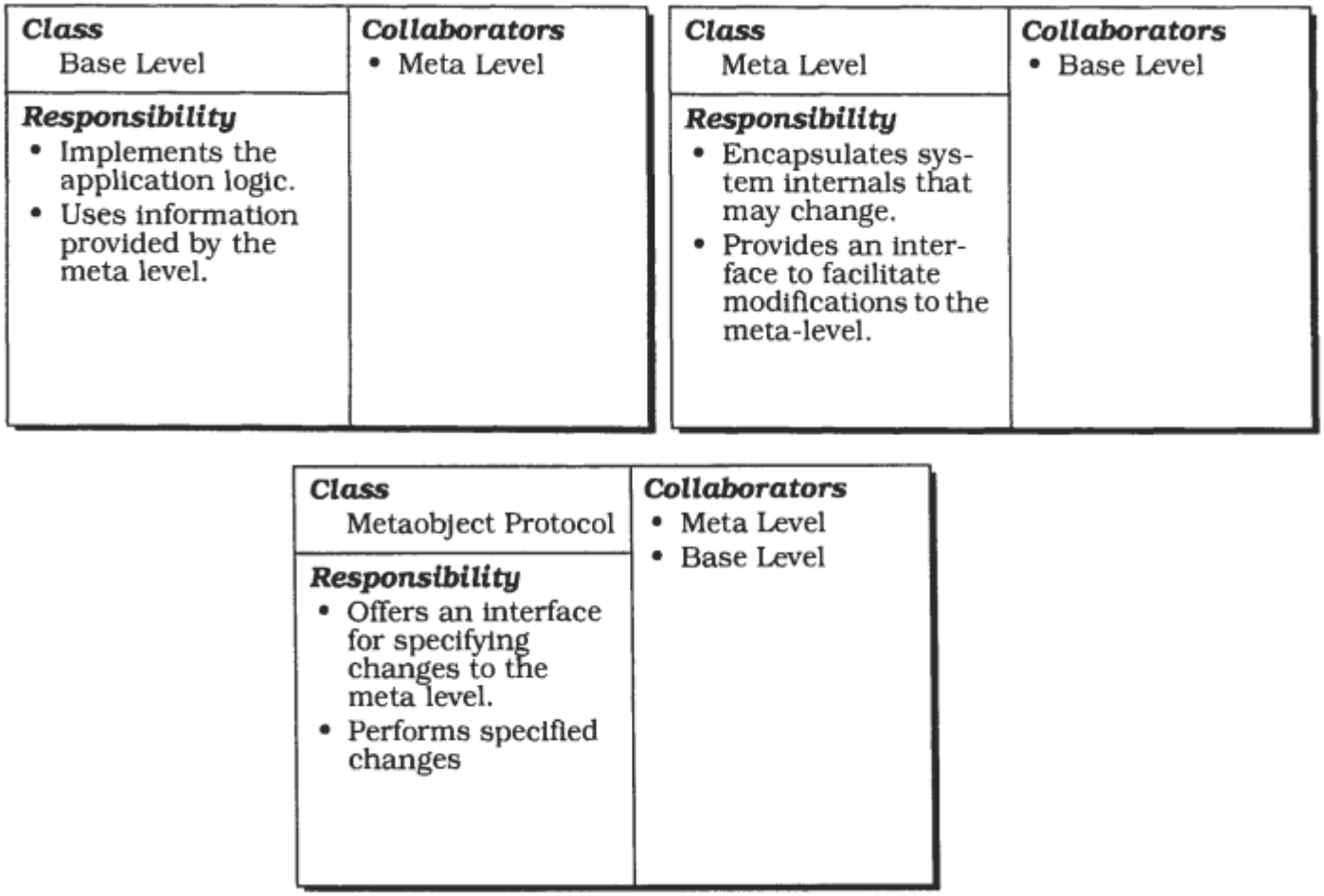
- Software systems evolve over time. They must be open to modifications in response to changing technology and requirements
- Changing software is tedious, error prone, and often expensive
- Adaptable software systems usually have a complex inner
- structure.



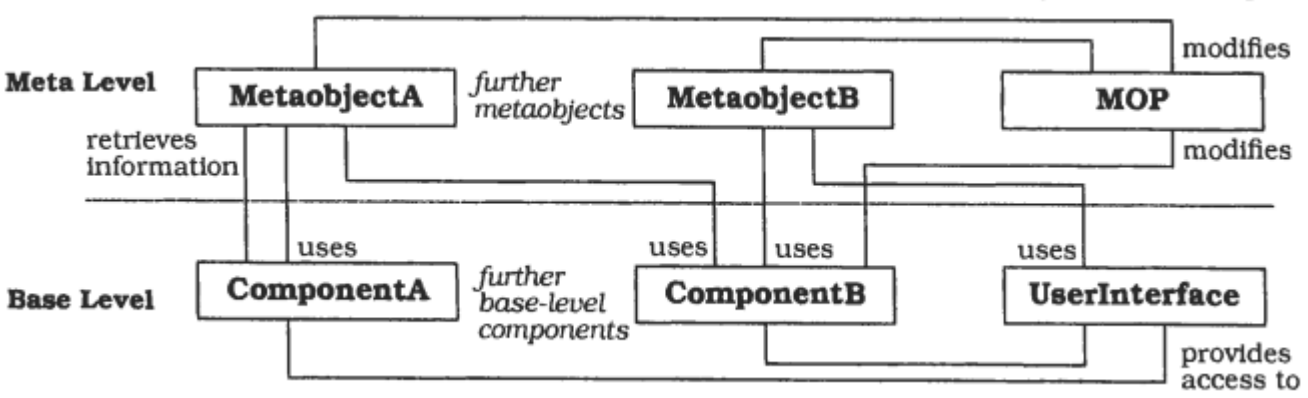
Solution

- Make the software self-aware, and make selected aspects of its structure and behavior accessible for adaptation and change.
- An architecture which is splited over two parts: a meta level
- and a base level.
- The meta level provides a self-representation of the software to give it knowledge of its own structure and behavior, and consists of so-called ***metaobjects.***
- The base level defines the application logic
- **An** interface is specified for manipulating the metaobjects. It is called the ***metaobject protocol*** (MOP)
- Persistence component located at base level

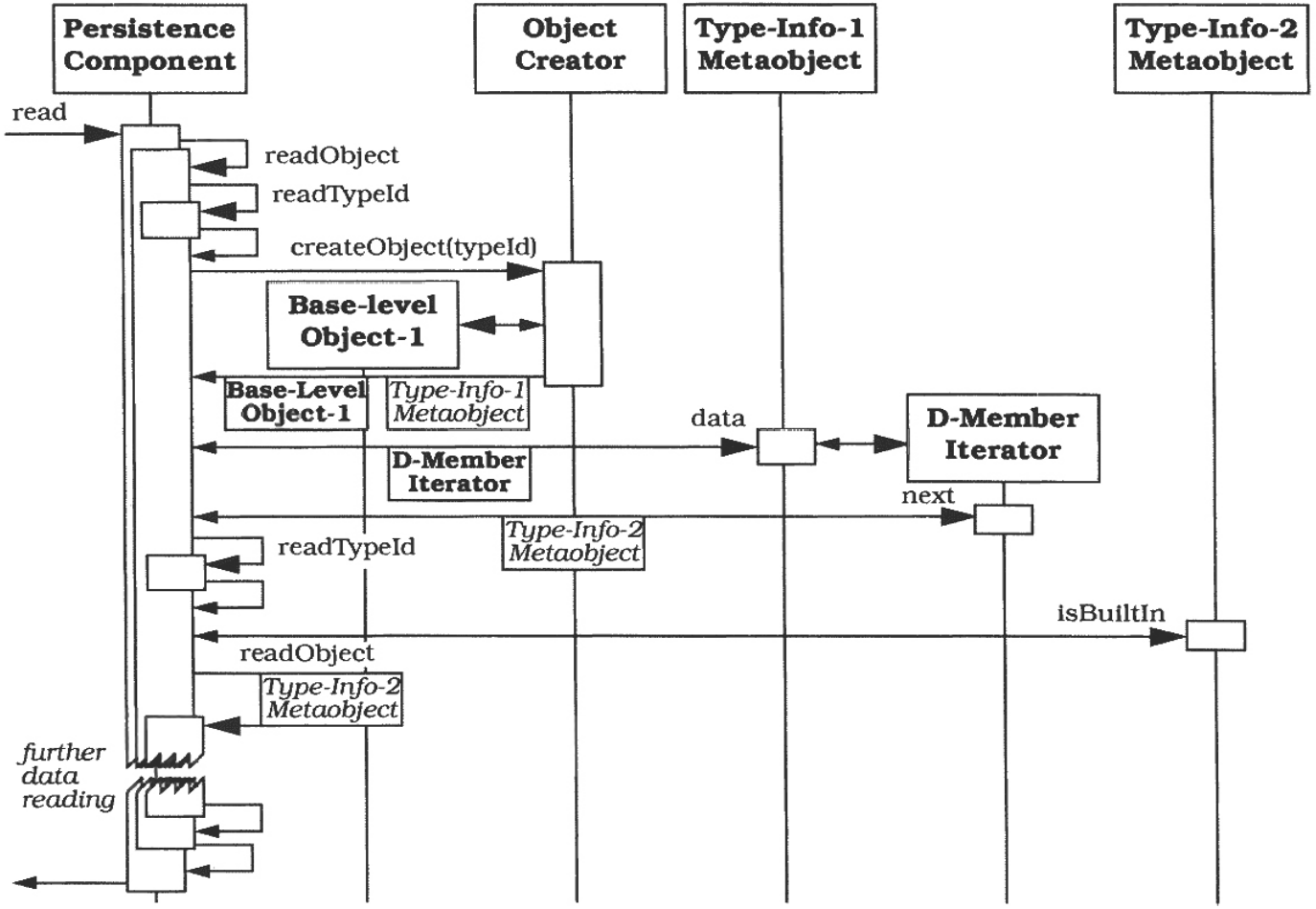
Structure


Class
Base Level
Collaborators
• Meta Level
Responsibility
• Implements the application logic.
• Uses information provided by the meta level.
Class
Meta Level
Collaborators
• Base Level
Responsibility
• Encapsulates system internals that may change.
• Provides an interface to facilitate modifications to the meta-level.
Class
Metaobject Protocol
Collaborators
• Meta Level
• Base Level
Responsibility
• Offers an interface for specifying changes to the meta level.
• Performs specified changes

Contd.

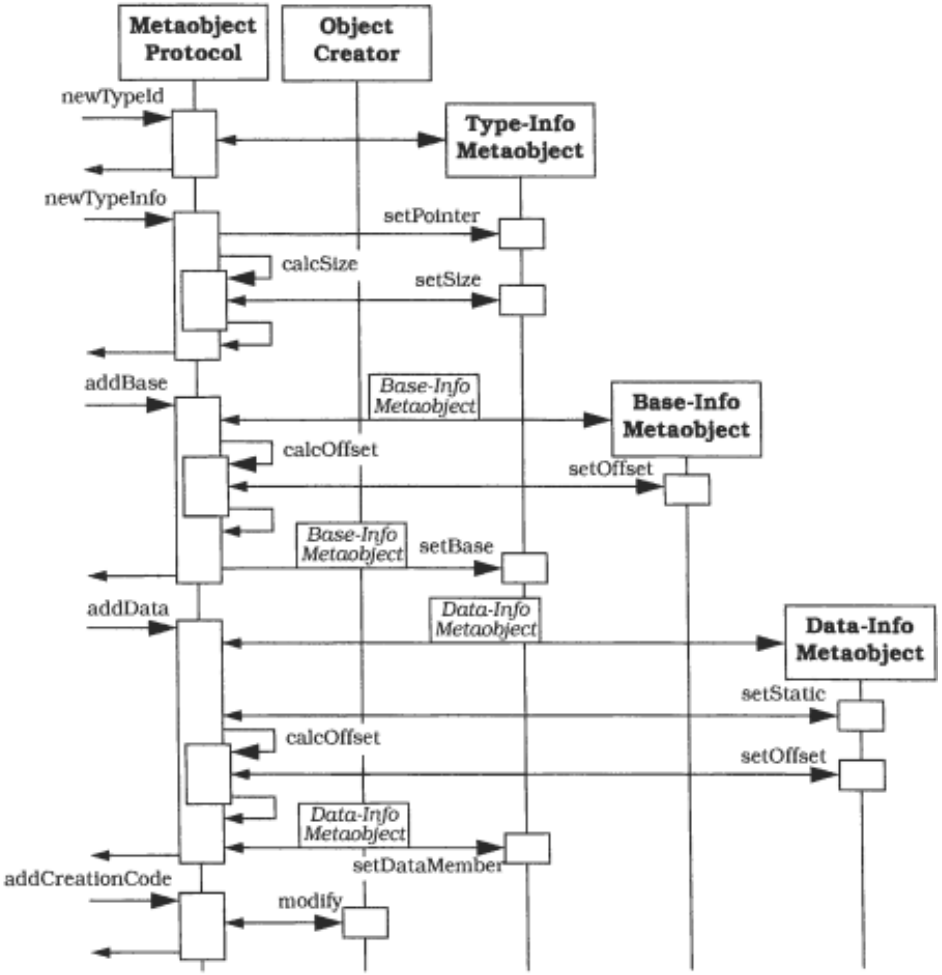

Meta Level
MetaobjectA
further metaobjects
MetaobjectB
MOP
modifies
modifies
retrieves information
uses
uses
uses
uses
Base Level
ComponentA
further base-level components
ComponentB
UserInterface
provides access to

Implementation



Collaboration between base level and meta level

Contd.



Meta object protocol

Applications

- MIP - run-time type information system for C++.
- Pgen - persistence component for C++
- NEDIS - includes a meta level called run-time data dictionary. Etc.

Pros & Cons

- No explicit modification of source code.
- Changing a software system is easy
- Support for many kinds of change
- Modifications at the meta level may cause damage
- Increased number of components
- Lower efficiency
- Not all languages support reflection