Design Issue #8
User Operations

- Many possible operations
  - Document operations: create, open, save, print a document
  - Editing operations: select, copy, cut, paste, undo, redo
  - Formatting operations: text formatting, character formatting
  - Miscellaneous operations: context sensitive help
- Different interfaces for these operations
  - Different Look-and-Feel
  - Different Windowing Systems
  - Different Access Points (menu, shortcut key, context menu)
- Independence from the UI
  - UI is a possible trigger, but not the only one;
  - *What* is done should not depend on the UI

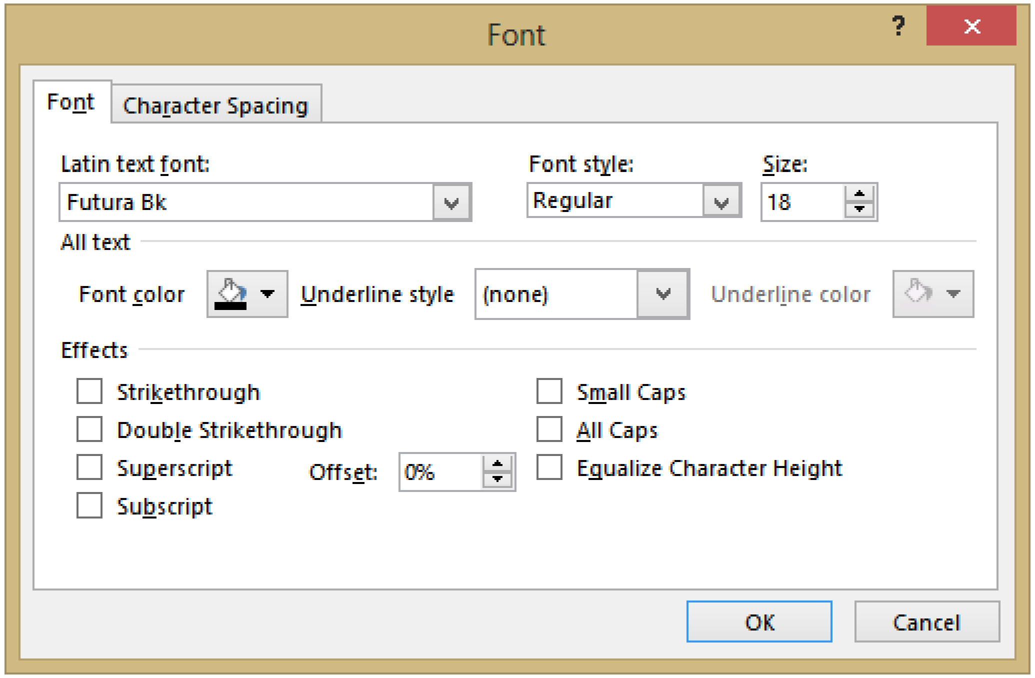
Design Issue #8

User Operations

- Many possible operations
  - Document operations: create, open, save, print a document
  - Editing operations: select, copy, cut, paste, undo, redo
  - Formatting operations: text formatting**, character formatting**
  - Miscellaneous operations: context sensitive help
- Different interfaces for these operations
  - Different Look-and-Feel
  - Different Windowing Systems
  - Different Access Points (menu, shortcut key, context menu)
- Independence from the UI
  - UI is a possible trigger, but not the only one;
  - *What* is done should not depend on the UI

Design Issue #8

User Operations


Font

Font | Character Spacing

Latin text font: Futura Bk

Font style: Regular

Size: 18

All text

Font color

Underline style: (none)

Underline color

Effects

- [ ] Strikethrough
- [ ] Double Strikethrough
- [ ] Superscript Offset: 0%
- [ ] Subscript
- [ ] Small Caps
- [ ] All Caps
- [ ] Equalize Character Height

OK Cancel

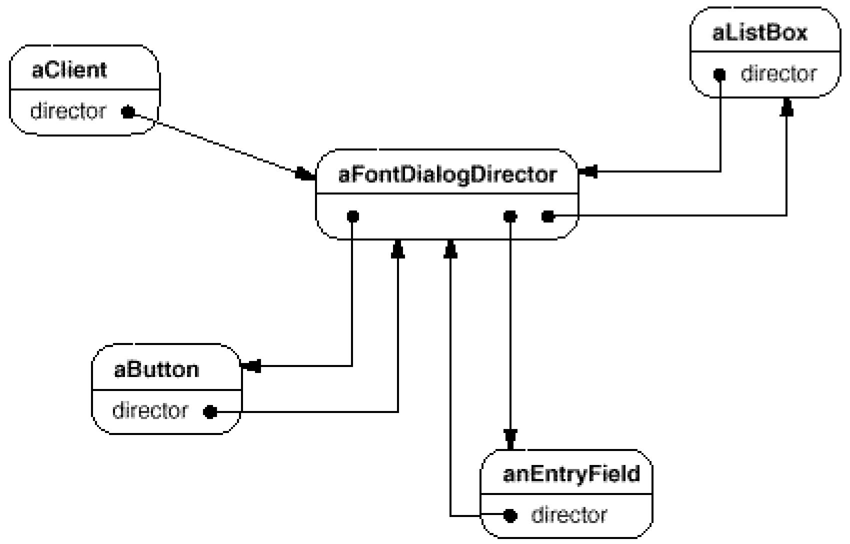
Design Issue #8

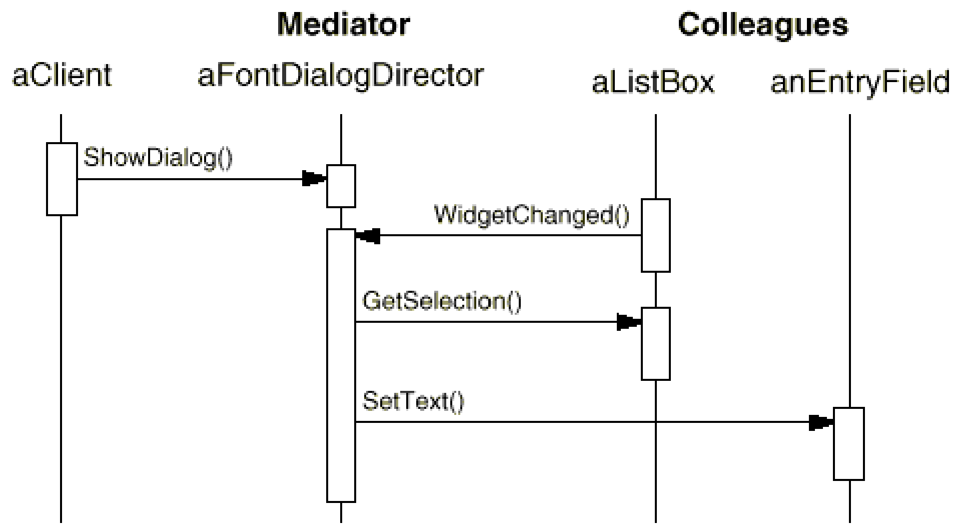
User Operations

Team exercise: Implement the collaborations behind Character formatting dialog.

Design Issue #8

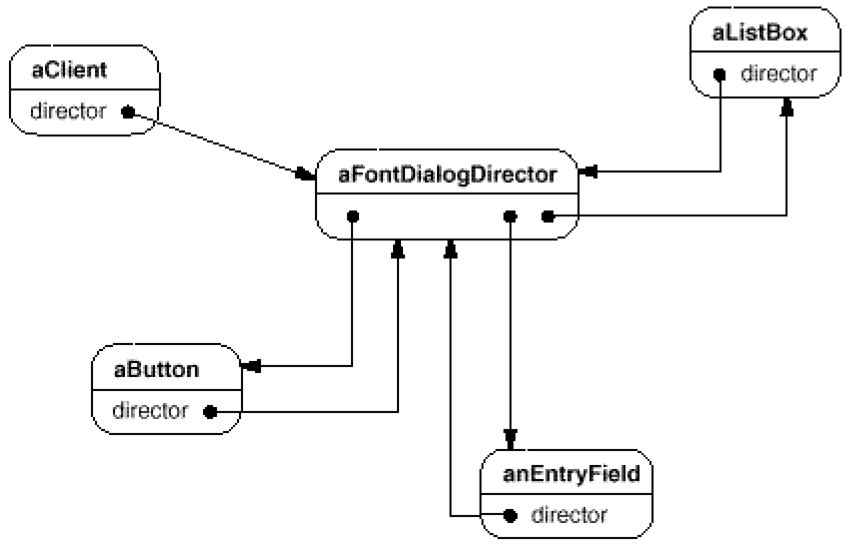
User Operations





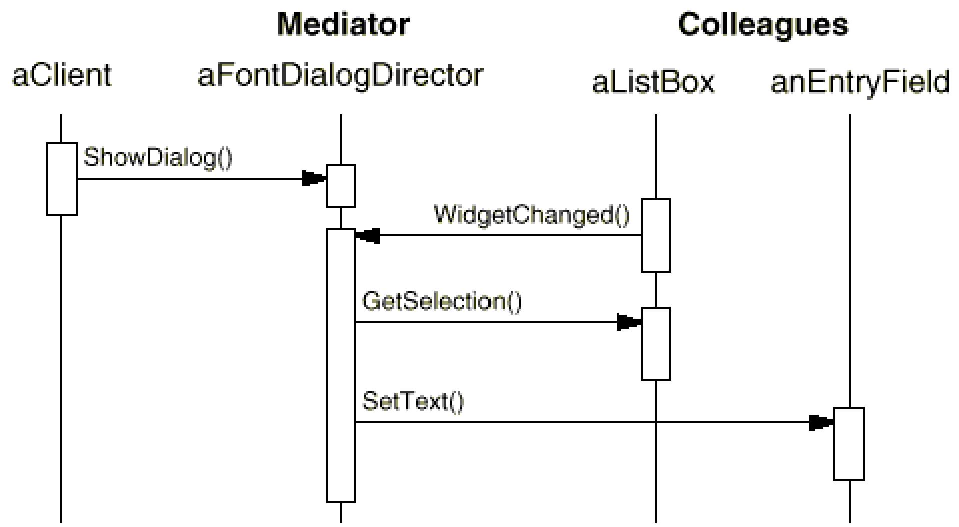
Design Issue #8

User Operations



Just met the

Mediator pattern

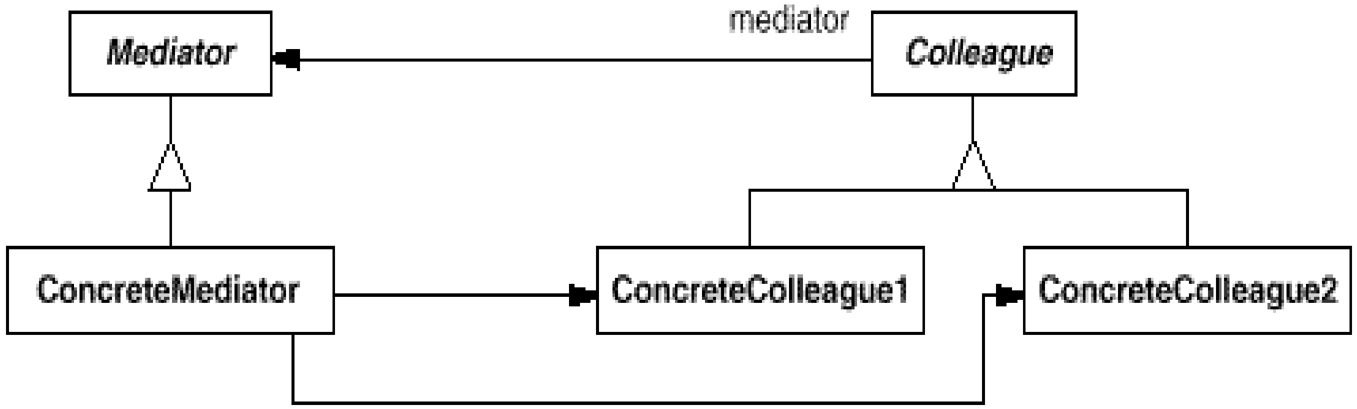


Design Issue #8

User Operations

Define an object that encapsulates how a set of objects interact.

Mediator promotes **loose coupling** by keeping objects from referring to each other explicitly, and it lets you **vary** their **interaction independently**.



limits subclassing

simplifies object protocols

decouples colleagues

abstracts how objects cooperate

centralizes control

Design Issue #8

User Operations

Applicability:

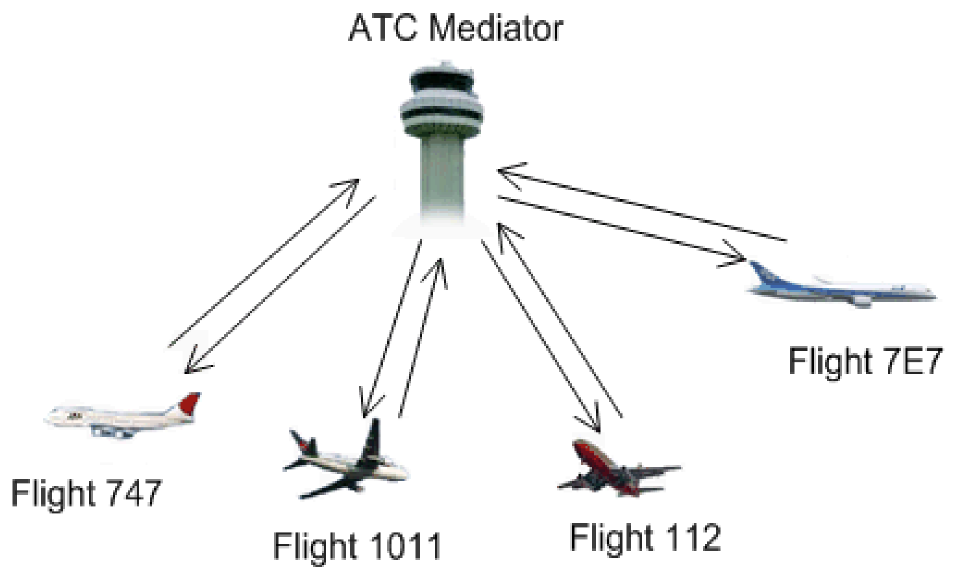
- a set of objects communicate in well-defined but complex ways
- reusing an object is difficult because it refers to and communicates with many other objects
- a behavior that's distributed between several classes should be customizable without a lot of subclassing

Remarks:

- Omitting the abstract Mediator class.
- Colleague-Mediator communication.

Design Issue #8

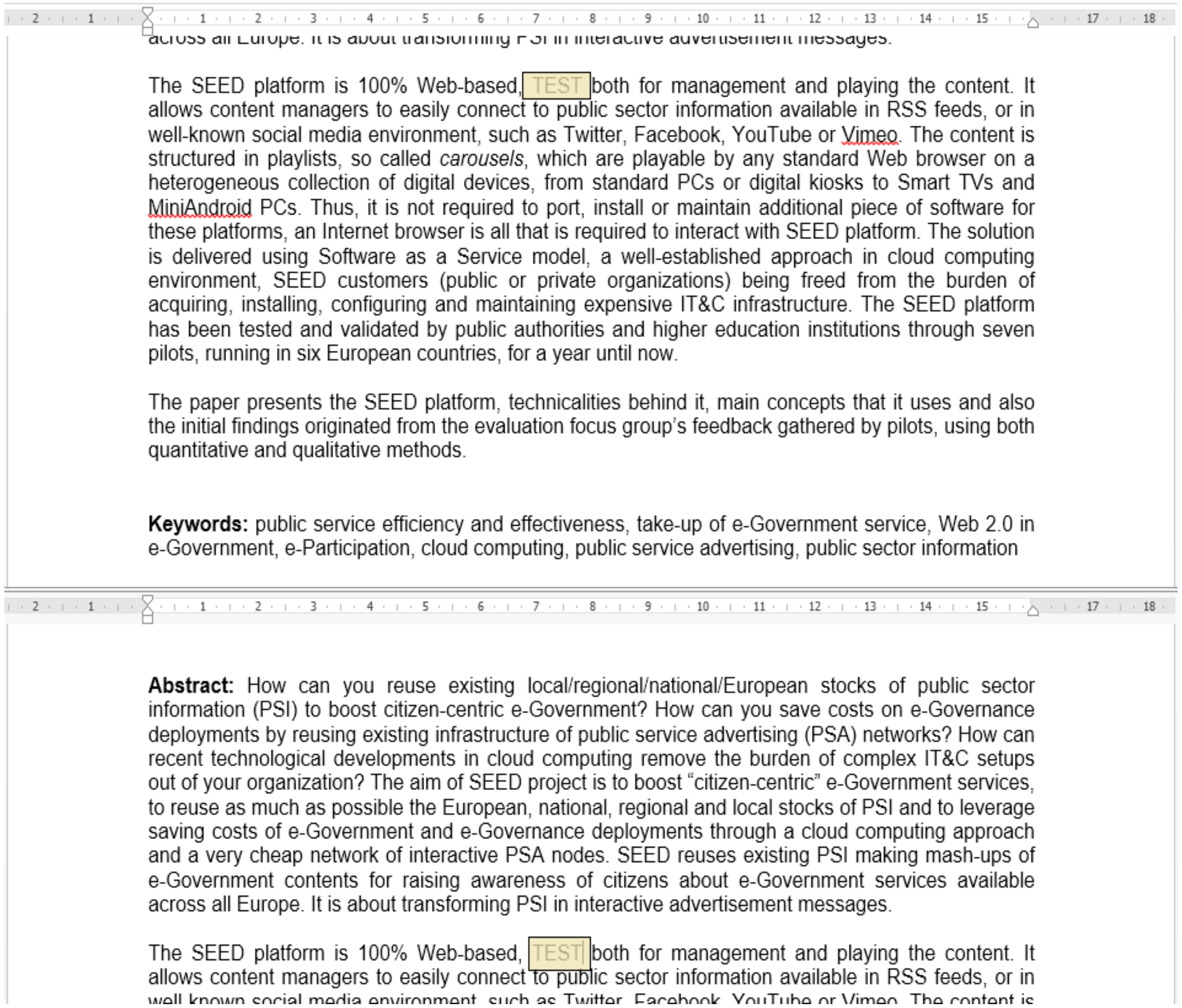
User Operations



- The control tower at a controlled airport demonstrates this pattern very well.
- The pilots of the planes approaching or departing the terminal area communicate with the tower rather than explicitly communicating with one another.
- The constraints on who can take off or land are enforced by the tower.
- It is important to note that the tower does not control the whole flight.
- It exists only to enforce constraints in the terminal area.

Design Issue #8
User Operations

- Splitting the view


across all Europe. It is about transforming PSI in interactive advertisement messages.

The SEED platform is 100% Web-based, TEST both for management and playing the content. It allows content managers to easily connect to public sector information available in RSS feeds, or in well-known social media environment, such as Twitter, Facebook, YouTube or Vimeo. The content is structured in playlists, so called *carousels*, which are playable by any standard Web browser on a heterogeneous collection of digital devices, from standard PCs or digital kiosks to Smart TVs and MiniAndroid PCs. Thus, it is not required to port, install or maintain additional piece of software for these platforms, an Internet browser is all that is required to interact with SEED platform. The solution is delivered using Software as a Service model, a well-established approach in cloud computing environment, SEED customers (public or private organizations) being freed from the burden of acquiring, installing, configuring and maintaining expensive IT&C infrastructure. The SEED platform has been tested and validated by public authorities and higher education institutions through seven pilots, running in six European countries, for a year until now.

The paper presents the SEED platform, technicalities behind it, main concepts that it uses and also the initial findings originated from the evaluation focus group's feedback gathered by pilots, using both quantitative and qualitative methods.

Keywords: public service efficiency and effectiveness, take-up of e-Government service, Web 2.0 in e-Government, e-Participation, cloud computing, public service advertising, public sector information

Abstract: How can you reuse existing local/regional/national/European stocks of public sector information (PSI) to boost citizen-centric e-Government? How can you save costs on e-Governance deployments by reusing existing infrastructure of public service advertising (PSA) networks? How can recent technological developments in cloud computing remove the burden of complex IT&C setups out of your organization? The aim of SEED project is to boost "citizen-centric" e-Government services, to reuse as much as possible the European, national, regional and local stocks of PSI and to leverage saving costs of e-Government and e-Governance deployments through a cloud computing approach and a very cheap network of interactive PSA nodes. SEED reuses existing PSI making mash-ups of e-Government contents for raising awareness of citizens about e-Government services available across all Europe. It is about transforming PSI in interactive advertisement messages.

The SEED platform is 100% Web-based, TEST both for management and playing the content. It allows content managers to easily connect to public sector information available in RSS feeds, or in well-known social media environment, such as Twitter, Facebook, YouTube or Vimeo. The content is

Design Issue #8

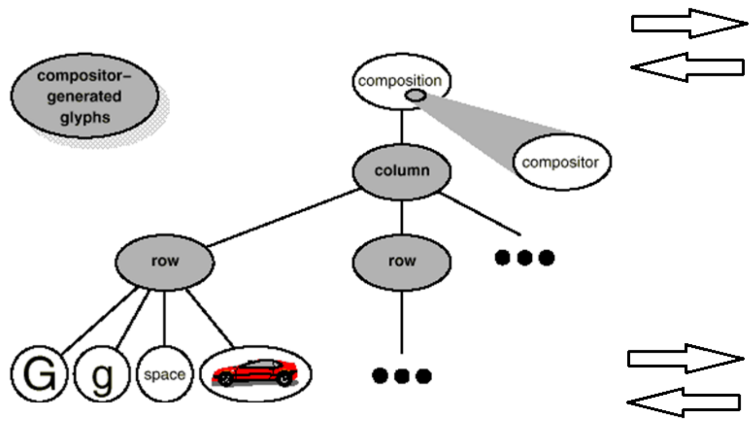
User Operations

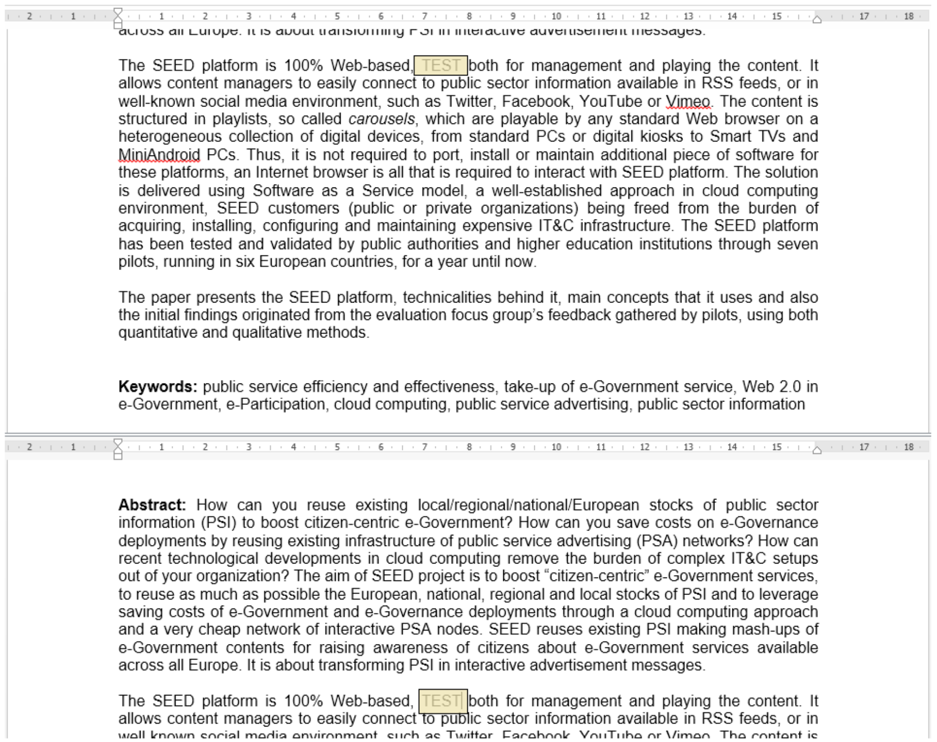
Team exercise: Open one document in 2 views and maintain their state consistent

Design Issue #8

User Operations

- Define a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.
- Maintain consistency between dependent objects.




across all Europe. It is about transforming PSI in interactive advertisement messages.

The SEED platform is 100% Web-based, TEST both for management and playing the content. It allows content managers to easily connect to public sector information available in RSS feeds, or in well-known social media environment, such as Twitter, Facebook, YouTube or Vimeo. The content is structured in playlists, so called *carousels*, which are playable by any standard Web browser on a heterogeneous collection of digital devices, from standard PCs or digital kiosks to Smart TVs and MiniAndroid PCs. Thus, it is not required to port, install or maintain additional piece of software for these platforms, an Internet browser is all that is required to interact with SEED platform. The solution is delivered using Software as a Service model, a well-established approach in cloud computing environment, SEED customers (public or private organizations) being freed from the burden of acquiring, installing, configuring and maintaining expensive IT&C infrastructure. The SEED platform has been tested and validated by public authorities and higher education institutions through seven pilots, running in six European countries, for a year until now.

The paper presents the SEED platform, technicalities behind it, main concepts that it uses and also the initial findings originated from the evaluation focus group's feedback gathered by pilots, using both quantitative and qualitative methods.

Keywords: public service efficiency and effectiveness, take-up of e-Government service, Web 2.0 in e-Government, e-Participation, cloud computing, public service advertising, public sector information

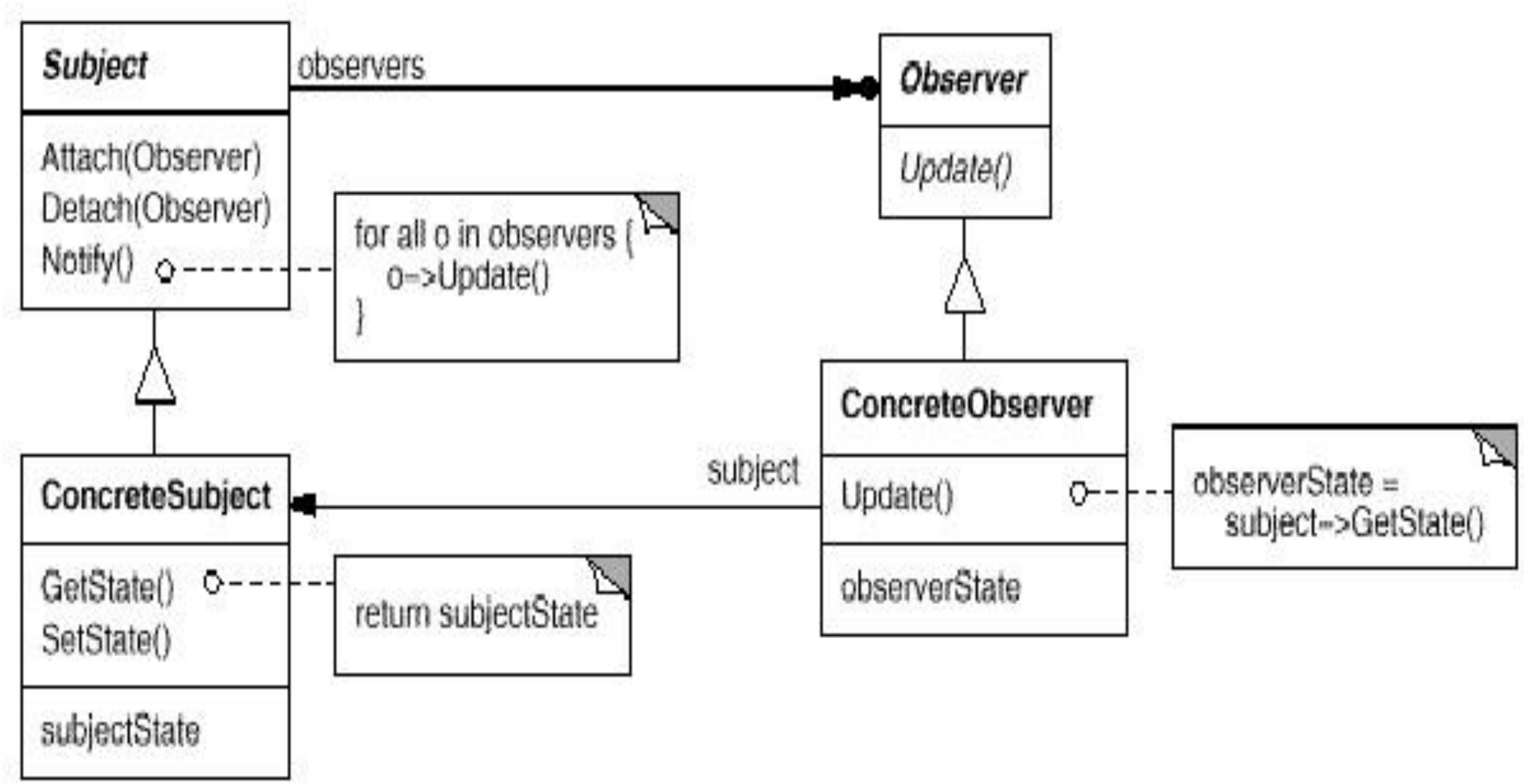
Abstract: How can you reuse existing local/regional/national/European stocks of public sector information (PSI) to boost citizen-centric e-Government? How can you save costs on e-Governance deployments by reusing existing infrastructure of public service advertising (PSA) networks? How can recent technological developments in cloud computing remove the burden of complex IT&C setups out of your organization? The aim of SEED project is to boost "citizen-centric" e-Government services, to reuse as much as possible the European, national, regional and local stocks of PSI and to leverage saving costs of e-Government and e-Governance deployments through a cloud computing approach and a very cheap network of interactive PSA nodes. SEED reuses existing PSI making mash-ups of e-Government contents for raising awareness of citizens about e-Government services available across all Europe. It is about transforming PSI in interactive advertisement messages.

The SEED platform is 100% Web-based, TEST both for management and playing the content. It allows content managers to easily connect to public sector information available in RSS feeds, or in well known social media environment, such as Twitter, Facebook, YouTube or Vimeo. The content is

Observer pattern

Design Issue #8

User Operations



Design Issue #8

User Operations

Applicability:

- When an abstraction has two aspects, one dependent on the other. Encapsulating these aspects in separate objects lets you vary and reuse them independently.

- When a change to one object requires changing others, and you don't know how many objects need to be changed.

- When an object should be able to notify other objects without making assumptions about who these objects are. In other words, you don't want these objects tightly coupled.

Design Issue #8

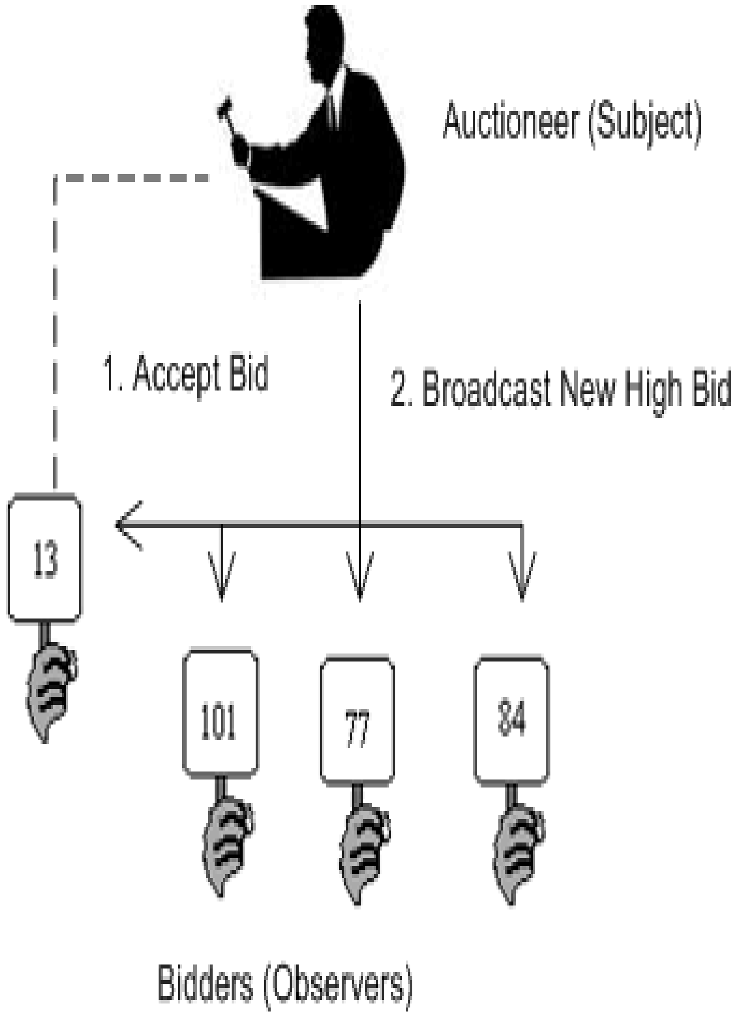
User Operations

- Abstract coupling between Subject and Observer
- Support for broadcast communication
- Unexpected updates
- Who triggers the updates?
- Observing more than one subject
- Mapping objects to their observers
- Ensure self-consistency of subject state before notifications

Design Issue #8
User Operations

The auctioneer (subject) observes new bids offered by bidders (observers).

The new price is then broadcasted to all bidders.



Summary

Covered patterns

- Very good example: 21 out of 23 patterns (91%) described in Design Patterns: Elements of Reusable Object-Oriented Software were covered within this example:

- **Structural**: Composite, Flyweight, Proxy, Bridge, Adapter

- **Behavioral**: Strategy, State, Decorator, Template Method, Command, Memento, Iterator, Visitor, Mediator, Observer, Chain of Responsibility

- **Creational**: Singleton, Abstract Factory, Factory Method, Prototype, Builder