Towards Formal Specification of PaperLess Office Systems

Abstract—The Center for Information and Communication Technology Services of Gadjah Mada University runs a kind of formal office communication system called PLO (stands for PaperLess Office). Developed to serve ad hoc needs of the center, the system evolves for several years without proper documentation. Several attempt store write the system failed due to the lack of the documentations. Judging from the fact that it has been serving the institution well for several years, the system can be treated as a user requirement. A formal specification is written based on it.

Keywords—formal specification; paperless office; information system

I. INTRODUCTION

Since 2004, the Center for Information and Communication Technology Services of Gadjah Mada University runs an information system called PLO (stands for PaperLess Office). In terms of software engineering, the system serves as a rapid prototype of an office communication system.

Historically PLO replaced the mailing-list based office communication management which did not work well due to it’s exponential growth as described in [1]. PLO project worked so well in the office such that it not only reduced annual paper use down to 10% of the previous one but also eliminated the need of conducting office meetings. Its communication threads are so well organized such that they also serve as minutes of meetings.

Being operational while developed incrementally by a single person, PLO’s modules have never been inspected formally for their security, functionality and efficiency. Lacking documentations, numerous attempts to rewrite the codes failed.

In 2011, the government of Indonesia issue an official regulation for system conducting government electronic communications [2]. The formal specification of the paperless office is also needed to examine it’s compliance to the regulation.

To build industrial grade PLO system, we need formal specification of the system. This document is the first attempt to document the PLO formally based on the operation PLO system as the prototype. Z specification language is utilized for its simplicity yet rich enough to capture the dynamic nature of office data [3].

II. STATE

In this section, we specify states of PLO elements and conditions among them that must be hold all the time. The conditions address issues of formal condition of data integrity as well as security requirements.

A. Basic Types

There are 4 basic types in PLO. They are:

\[ \text{PERSON, MEMO, MEDIA, GROUP} \]

\[ \text{LOCATION} \]

\[ \text{DateTime} \]

B. The User

Users are PERSON who have access to the system.

\[ \text{User}:: \text{\#PERSON} \]

To simplify the specification, usual attributes of person such as names, unique identifiers, sexes, and dates of birth are
omitted. Initially, there is one user, called admin which has the authority of setting up the operation of the system.

\[\text{InitUser} \quad \Delta \text{User} \quad \text{admin} : \text{PERSON} \]

user' = \{admin\}

C. Message

Messages are MEMO written by the users at specific time.

\[\text{Message} \quad \Delta \text{Message} \quad \text{memo}' = \emptyset\]

A MEMO may be associated with a set of attached MEDIA files. A MEDIA can be attached to asset of MEMOs.

\[\text{Attachment} \quad \Delta \text{Message} \quad \text{file} : \text{MEMO} \leftrightarrow \text{MEDIA} \]

Initially, the memo set is empty.

\[\text{InitMessage} \quad \Delta \text{Message} \quad \text{memo}' = \emptyset\]

D. Grouping

Grouping is used to define groups of users. The purpose of specifying groups is to make it shorter list of memo’s target readers. The grouping is also used to control access.

\[\text{Group} \quad \Delta \text{User} \quad \text{group} : \text{PERSON} \leftrightarrow \text{GROUP} \]

\[\text{member} : \text{GROUP} \leftrightarrow \text{PERSON} \]

Initially, there is one group of users having the authority of setting up the operation of the user.

\[\text{InitGroup} \quad \Delta \text{Group} \quad \text{ads} : \text{GROUP} \]

\[\text{group'} = \{\text{ads}\}\]

group' = \{ads\}

E. Target

Each memo has at least one target of user which can be specified directly by mentioning the name of the user or indirectly via group membership by mentioning a group.

\[\text{Target} \quad \Delta \text{Message} \quad \text{userTarget} : \text{GROUP} \leftrightarrow \text{PERSON} \]

\[\text{groupTarget} : \text{GROUP} \leftrightarrow \text{GROUP} \]

\[\text{allUserTarget} : \text{PERSON} \leftrightarrow \text{PERSON} \]

Users in a memo’s target may access media attached to it.

\[\text{MediaAccess} \quad \Delta \text{Target} \quad \text{mediaAccess} : \text{PERSON} \leftrightarrow \text{MEDIA} \]

F. Comments

Comments are memos which follow with other memos. Comments are specified by defining global isFollowing function.

\[\text{Comment} \quad \Delta \text{Target} \quad \text{isFollowing} : \text{MEMO} \leftrightarrow \text{MEMO} \]

\[\text{dom isFollowing} \subseteq \text{memo}\]
ran isFollowing ∈ memo
∀m1,m2 : memo • m1 = isFollowing(m2) ⇒
postedTime(m1) > postedTime(m2) ∧
allUserTarget(m1) = allUserTarget(m2)

G. Read

We specify read to mark time when users access memos.

Read
Target : PERSON × MEMO ⇒ DateTime

isRead : PERSON × MEMO ⇒ DateTime

dom isRead ⊆ user × memo
∀ p : user • m : memo •
   p ∈ allUserTarget(m) ∧
   (p,m) ∈ dom isRead ⇒
   postedTime(m) < isRead(p,m)

III. OPERATION

PLO operations are procedures of querying the states which can modify them.

A. Listing Memos

The global function listMemo returns all main memos users have access to. Main memos are memos which are not comments. They do not follow any other memo. Listing memos does not change PLO’s states.

listMemo : PERSON ⇒ ∃ MEMO

null listMemo = user
∀ u : user •
   listMemo(u) =
   { (m : memo | u ∈ allUserTarget(m) ∧
       ¬ (∃ n : memo • (m,n) ∈ isFollowing))}

The return values of listMemo can be ordered by postedTime either descending or ascending.

B. Viewing a Memo

The operation of viewing a memo returns all comments following the memo.

viewMemo : PERSON × MEMO ⇒ ∃ Comment

null viewMemo = user
∀ u : user • m : memo •
   viewMemo(u,m) =
   { m : memo | u ∈ allUserTarget(m) ∧
       ¬ (∃ n : memo • (m,n) ∈ isFollowing) ∧
       ¬ (∃ c : Comment • (m,n) ∈ isFollowing)}

The reversion of viewing a memo returns all comments.

C. Notification

Notifications are sets of memos which have not been accessed by the users.

Notify
Target : PERSON ⇒ MEMO

dom Notify = user
∀ u : user • Notify(u) =
   { n : memo | u ∈ allUserTarget(n) ∧
       n ∈ allUserTarget(n) ∧
       ¬ (∃ c : Comment • (m,n) ∈ isFollowing) ∧
       ¬ (∃ c : Comment • (m,n) ∈ isFollowing)}

D. Adding a Memo

The AddMemo schema specifies the behavior of the system when the state of Message is modified by adding a new one. The new memo brings authorship and optional attachment and target reader properties with it.

AddMemo
Target : PERSON × MEMO ⇒ ∃ Comment

null AddMemo = user
∀ u : user • m : memo •
   AddMemo(u,m) =
   { n : memo | u ∈ allUserTarget(n) ∧
       n ∈ allUserTarget(n) ∧
       ¬ (∃ c : Comment • (m,n) ∈ isFollowing) ∧
       ¬ (∃ c : Comment • (m,n) ∈ isFollowing)}
E. Adding a Comment

A comment id a memo to follow another memo.

```
AddComment
AddMemo
\Delta Comment
parent? : MEMO

parent? \in memo
isFollowing' =
isFollowing \cup \{ \text{newMemo}? \mapsto parent? \}
```

IV. ADDITIONAL FEATURES

In addition to the basic features, there are several additional ones that are useful to the users. There are news, blogs, event calendar and expressions. News and events have additional attributes are formally specified while blogs and expression are not explicitly specified since they are merely subsets of news.

A. News

News are memos targeted to all users.

```
News
Target
news : \mathbb{P} MEMO

news \subseteq memo
\forall n : news \cdot userTarget(n) = user
```

B. Event Calendar

Calendar events are memos with additional information of locations, times and durations. The set of LOCATIONs are introduced to abstract locations where the events occur. There are two disjoint sets of LOCATIONs, location and xlocation; location is the set of LOCATION where events can occur at the same time and xlocation is the set of LOCATION where only one event can occur at the same time.

```
[LOCATION]
EventCalendar
Message
event : \mathbb{P} MEMO
occur : MEMO \leftrightarrow \text{DateTime}
duration : MEMO \leftrightarrow \mathbb{N}
location : MEMO \leftrightarrow \text{LOCATION}
xlocation : MEMO \leftrightarrow \text{LOCATION}

event \subseteq memo
dom occur = event
dom duration = event
dom location \cup \text{dom xlocation} = \text{event}
dom location \cap \text{dom xlocation} = \emptyset
\forall a, b : \text{dom xlocation} \cdot
\quad \text{occur}(a) + \text{duration}(a) \leq \text{occur}(b) \lor
\quad \text{occur}(b) + \text{duration}(b) \leq \text{occur}(a)
```

V. SECURITY ISSUES

PLO security relies on the traditional user id password login system. Logged user are in the log set which initially is empty. Password is a set of natural numbers assigned to the users.

A. Login

Login is a set of users authorized to access the system.

```
Login
\Delta Log

User
login : \mathbb{P} PERSON
password : PERSON \leftrightarrow \mathbb{N}

login \subseteq user
dom password = user
```

Initially login is empty.

```
InitLogin
\Delta Log

login' = \emptyset
```

Users log in by submitting their pair of userid and passwords.

```
Login
\Delta Log

userid? : \text{PERSON}
password? : \mathbb{N}

password? = \text{password(userid?)}
password' = \text{password} \cup \{ \text{userid}? \mapsto \text{password}? \}
```

B. Logged Operation

All operations should be logged. For example, listing memo should be logged. The global variable actor? is introduced as the one who operate the system.

```
LoggedOperation

LoggedMemoList

MemoList
Log

actor? : \text{PERSON}
listing! : \mathbb{P} MEMO

actor? \in login
listing! = \text{listMemo}(actor?)
```

Person who add a comment have to be in the set of it’s parent memo

```
LoggedAddComment

AddComment
Log

actor? : \text{PERSON}
```
VI. CONCLUDING REMARKS

A formal specification of the PaperLess Office system has been written. Some organizational and security issues have been addressed in the specification. The formal specification has been fully implemented. The success of the system relies on the design which does not place the system as a resemblance the traditional office business as usual but places the system as the core engine to run the office electronically.

REFERENCES