IJMSIT

International Journal of Multidisciplinary Studies and Innovative Technologies

Volume : 5
Number: 2
Year: 2021
Pages: 155 - 166

Modelling Shared Co-Owned Data Flow in Online Social Networks by Formal Methods

Gulsum Akkuzu Kaya*, Benjamin Aziz [2]

[1]*Computer Engineering, Recep Tayyip Erdogan University, Rize, TURKEY*
[2]*School of Computing, University of Portsmouth, Portsmouth, United Kingdom*
**Corresponding author: gulsum.akkuzukaya@erdogan.edu.tr*

Abstract **–**Online social networks are common platforms for people to make connections and communicate to others. People are given a virtual space to share data either including only their own ids or including other users' ids. Data sharing sometimes cause privacy issues in online social networks because of inclusion of other users' ids. Researchers have studied on the privacy issues and these online platforms have taken measurements to preserve privacy leakage because of its inclusion on data. Users are not only allowed to share a content of data but also re-share a shared content. Re-sharing has also caused privacy issues in online social network platforms. Recently, Facebook has made an update on shared contents, in which permissions have been restricted based on groups. However, it has not solved the main issue since the proposed solution is a coarse-grained control not a fine-grained control on shared contents of data. This work introduces a fine-grained control flow on shared contents in which users' reputation and data sensitivity are used. To specify our proposed work's specifications and verify the proposed model, we used formal modelling. Formal analysis of this work is used to prove the applicability of the model and verification of the specifications.

Keywords –Data Flow, Data Sensitivity, Event-B, Formal Modelling, Online Social Networks

I. INTRODUCTION

Online social networks (OSNs) play a crucial role in people's lives. OSNs are the virtual platforms where people communicate and interact with others via sharing contents of data such as photo, video, text, and audio. The definition of OSNs can be varied depending on what the network itself offers, however, one thing is common for all OSNs which is providing an environment where users post a profile and communicate with others via sharing data. Each member of OSNs is provided a virtual space to post contents of data and keep their information in. Moreover, members are allowed to post contents of data to their own spaces and other users' spaces. For this reason, the contents of data can be grouped into two classes in OSNs, the first one is the single-handed content which is related to the only one user, who shares the content. The second type of the content is co-owned data which has more than one user information on. It is easy to decide the sensitivity of the single-owned contents since there is one user to make decision on sensitivity level of data. However, valuing the co-owned content's sensitivity needs all users' opinions on it [1]. This is because co-owned data may not be sensitive to a user while it might be highly sensitive to other user whose information on it.

Most users who share sensitive co-owned contents in OSNs are unaware of leaking other users' privacy. In OSNs, data is usually encrypted with attribute-based encryption, any user could retrieve data, any member can decrypt the data. This may not be a problem for the first targeted group, however, when data is relieved by the members of the first targeted group it may flow to members whom is not meant to access the data. Controlling such information is of great concern. Therefore, flow control mechanisms are essential for protecting the shared contents from unwanted users and preventing users' privacy leakage in OSNs.

Facebook has recently updated their shared contents of data policies [2] with a very coarse-grained adjustment. The updates is mainly based on the groups in the Facebook, for example, if a user shares a photo with only his friends (i.e. first targeted group), then the shared content is not flown to the further users. Itis an important attempt to protect users' privacy on the shared contents and/or controlling a shared content flow. However, it is not a complete solution since itis not a fine-grained adjustment. In other words, it is a group based solution not a singular user based which means not a fine-grained based adjustment. Also this coarse-grained adjustment does not give an appropriate service to OSNs main aims, this is because OSNs have been built to give access to shared data as many as possible. To solve these issues in OSNs platforms, we propose a fine-grained solution in which conditions are on a content sensitivity value and a user's reputation value.

Use of users' reputation in a co-owned data sharing process in OSNs was introduced in [4]. They proposed a new concept of OSNs in which each user has a reputation value and each co-owned data has the sensitivity value. We propose a control flow of a shared co-owned data with the usage of users' reputation values and co-owned data sensitivity value. To do that, we use Event-B [7], the aim is to keep high sensitive co-owned data in a secure sharing track, which means high sensitive data should not be shared with low reputed users. We use Event-B [8] for formal modelling of the flow control. Event-B is a formal method for modelling and analyzing system. It is used to model and develop systems based on the conditions. Key features of Event-B are the use of set theory

as a modelling notation, the use of refinement to represent systems at different abstraction levels and the use of mathematical proof to verify consistency between refinement levels.

The rest of the paper is organised as follows. Section 2 introduces similar works from the literature of formal modelling. The problem statement is high-lighted in Section 3 as well as the contribution of this work is given. In Section4 we given an overview of the Event-B syntax. We then present our proposed model and framework in Section 5. Section 6 and 7 present the implementation with Event-B formal modelling. Finally, the summary of this paper is provided in Section 8 with some future directions.

II. RELATED WORKS

The modelling and analysis of OSNs is not a new idea, it has been under-researched in many aspects. OSNs' formalization and modelling are usually done with the graph theory [9]. OSNs are considered as a set of nodes and edges that tie one node to another, nodes and edges are used to define OSNs [10]. Analysing the trace of data flow among nodes is linked to relationships. If two nodes have relationship, then the accessibility of the content is allowed [11, 20, 12, 13]. There are various proposed models for information flow control in OSNs in which the trust values between nodes are used [14, 19]. Akkuzu et al. [3] have recently introduced a new approach for secure data sharing processes in OSNs, they suggest to use not only users' trust values but also use their reputation values. Another proposed method for controlling information flow is group-centric models in which users' authorization in a group membership is used [15]. They used super distribution (SD) and micro distribution (MD)for providing a secure data sharing environment. Authors in [16] introduced anew model for controlling information flow in OSNs with mutual distrust and decentralised authority. A new OSN was introduced by Baden et al. [17] where users decide who can have access to their information.

Using formal methods for verifying developed systems has not been a new idea in the literature, researchers have used formal languages either to verify their proposed system or analyse proposed works more closely. Souri and Norouzi discussed the advantages of formal modelling usage to analyse any pro-posed systems' specifications and requirements in a survey research paper [24].Analysis of robotic systems' specifications and verification with formal modelling was discussed by Luckcuck et al [25]. The importance of formal modeling a system could be understood with the above two survey research papers. Formal modelling has not only been discussed by researchers they but also have been used to verify researchers' proposed works. For instance, there are various research was carried out for improving security in cloud services by formal approaches in [23, 27]. Amato et al. introduced a formal modelling concept for improving security in cloud services [23]. The aim was to observe crucial specifications and requirements for more secure cloud services. Formal modelling was also used in [26] with the aim of checking correctness of a software system. Another area of using formal modelling is social network platforms, there are researchers who used formal methods in social network areas for specifications and verification [28,29,22]. For example, Vishwamitra et al. used formal approaches to specify the targeted people for shared content and verify the specified system. Another example of using formal modelling in social networks was done by Abdulrahman et al. for understanding formal verification requirements of retrieving information from social network system [30].

All above research works have shown that formal models provide an environment for specifying requirements of a proposed system and verifying requirements of a proposed work. In the light of this idea, we use formal modelling to verify the proposed approach of this work. In order to verify the proposed approach, we use Event-B formal language for analysis of system-level-modelling of this work's proposed approach.

III. PROBLEM STATEMENT AND SUMMARY OF CONTRIBUTIONS

Each data needs be owned by a user not only in the real life communications, but also in communication of OSNs. The owner is the person who uploads, creates, shares, and/or controls a data in OSNs. In OSNs, there are two types of data in terms of ownership features, one is the single-owned data and the other one is the co-owned data. The Main difference between these two types of data is the single-owned data includes only one user's information and/or ids on and the co-owned data is related to more than one user. If the shared data is a single-owned data, then all the responsibilities belong to the owner, however, if the data is a co-owned data, then each user, whose information on the shared data, has right to take responsibility. For example, they may want to know whom will access to the shared data [5]. All these specifications are related to the first targeted group of people for the shared data, for example, a user wants to share a sensitive photo with only his family. If so, the user can specify the targeted group with "family members" then shares the data. As it is above mentioned, if the data is shared with a small group, then others are not permitted to access the data in the current OSNs. Although it is a solution for some cases, it is not an accurate approach. Because, it is a coarse-grained solution in which conditions are group based. For instance, people in the targeted group are not allowed to flow the data to the next group and if someone is not in the group but he is meant to access the data, then the current solution does not help to solve such problems in any circumstances. In order to overcome such problems in OSNs, fine-grained data flow control is a need. We therefore propose this work in which users' reputation values and data sensitivity value are used for controlling data flow. We use formal modelling to define the specifications and to verify the proposed approach since formal modelling well defined and commonly used for specifying and verifying software systems [6].

IV. AN OVERVIEW ON EVENT-B SYNTAX

The aim of this section is to give an overview explanation on Event-B language syntax, following explanations are given with the use of the work in [18]as base. In Event-B, there are

two basic constructs context and machine. The static part of a model in Event-B is defined in the context part. And the dynamic part of a model in Event-B is defined in machine part. Machines and contexts have different relationships: a machine can see one or various contexts for a model. A machine can be refined by another machine. Moreover, a context can be extended by another one.

Carrier sets, constants, axioms, and *theorems* are defined in contexts section in an Event-B programme. A *machine M* contains *variables, invariants, theorems, events*, and *variants*. *Variables v* define the state of a machine in Event-B. Variables are constrained by *invariants I(v).* Any changes in states are described in events.

Each event composes of a guard G and an action S, where the guard necessary states for an event and the action describes how the variable evolve when an event occurs. An event might have local variables. In such cases the representation of guard and action for the event being occurred are as; guard G(t,v) and an action *S(t,v*) where t indicates the local variable and v stands for the variables defined in I(v). An event E can be specified with three following forms;

$$E \triangleq \textbf{begin any } t \textbf{ where } G(t,v) \textbf{ then } S(t,v) \textbf{ end}$$

$$E \triangleq \textbf{begin when } G(v) \textbf{ then } S(v) \textbf{ end}$$

$$E \triangleq \textbf{begin } S(v) \textbf{ end}$$

Figure 1: Event-B form

Event-B has simple mathematical language, such as integers or given sets that are specific to a model or are formed from the Cartesian product and power-set type constructors. The definition of relations and functions is done by combining those constructors. Event-B language is designed with basic mathematical concepts therefore set theory and logic are used for descriptions as same as any engineering disciplines. Event-B notations therefore are defined in the same way of the mathematics notations. Table 1 gives some of the math notations, Event-B notations, and definitions.

Table 1. Mathematical Notation and Event-B Notation

Math Notation	Event-B Notation	Definition
∈	:	Set membership
ℕ	NAT	Natural numbers
≤	<=	Less than or equal
⊤	true	Boolean true
⊥	false	Boolean false
⊆	<:	Subset or equal
⊂	<	Strict subset not equal
→	−−>	Denotes a total function
+−>	+−>	Denotes a partial function
Ø	{ }	Empty set
≠	/=	Not equal
|−>	| - >	Maps to

V. The Proposed Model

In this work, there three roles for users in a co-owned data sharing process, for example, a user might be the owner, co-owner, or accessor (i.e. viewer) in OSNs. Each role should have different permissions and/or actions in co-owned data sharing processes in OSNs. Therefore, roles and activities have been defined considering that which roles can be given to a user and which activities are related to which roles. Figure 1 presents the structure of activities with their associated roles.

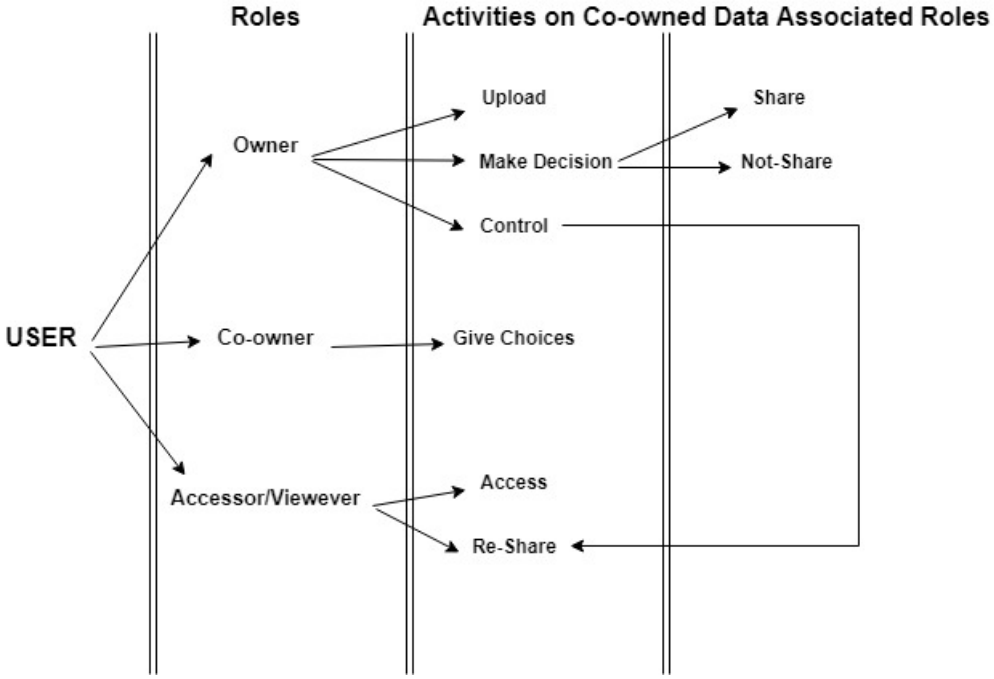

Fig. 1 Activities on co-owned Data Associated with a User's Role

USERS={u_1, u_2, u_3, …,u_k}, be a user set. The users are one of the main factors in OSNs since the main purpose of OSNs is to encourage users to be member in OSNs. Users are people who use OSNs for any purpose, however, a user might have different roles in different data sharing processes in OSNs. For instance, a user might be the owner of a content of data in OSNs, a viewer for another content of data, or a co-owner for the content of data. The term user covers all above mentioned cases.

DATA={d_1,d_2,d_3, …,d_l} be the set of contents of data shared in OSNs. The content of data can either be owned by only one user (i.e. single-owned data) or by several users (*i.e. co-owned data)*. Here, owning refers to the number of users' id on the content of data. If a content is owned by at least two users, then the content is called co-owned data.

ROLES={owner, co-owner, viewer/ accessor} be set of roles associated to users in the data sharing process. In OSNs, a user might become an owner for a shared content while he was a viewer for the same content before. In such a case, the content might be revealed to users who were not allowed by the first owner of the content. In order to cover this gap, we introduce a new activity control, where the first owner can specify following viewers/ accessors for the shared content. In this way, controlling the shared contents can be done in OSNs which is a way to preserve co-owners' privacy for the future flow of co-owned data.

ACTIVITIES={upload, take-decision, share/ not share, give-choices, access, re-share, control re-share} be the set of activities in OSNs related the roles associated to users in a data sharing process. In Figure 1, relationships between activities and roles have been given. In this work, the focus is on the association between activity re-share with the accessor/ viewer role and the control with owner role.

PERMISSIONS(Re-Share)=allow and deny be a set of permissions that demonstrates the first user, who shares the content of data, decided whether to control the flow of shared data or not. Re-share refers to the permission given to the first targeted group' members by the owner. The flow of shared data can be controlled with specifying permissions in the beginning of a data sharing sequence in OSNs. To do so, OSNs need more consideration on functions (*i.e. events*).

REPUTATION: be a set of integer numbers. We have given the models for users' reputation in OSNs. We now assume that each user is given a reputation value in OSNs. It is aforementioned that the reputation value is a dynamic value. It changes with respect to users' behaviours in a co-owned data sharing process. It increases if a user behaves in a good way, i.e. respecting co-owners' decisions. Sharing a co-owned content causes increment on the reputation value. Bad behaviour causes decrease in reputation value.

$$reputed[u] = i$$
$$where\ i\ \in \mathbb{R}$$
$$reputation[i] \in [0, \dots, \mathbb{R}]$$

DATA SENSITIVITY: be a set of integer numbers where the numbers range from 0 to 10. We have explained the model for the data sensitivity value and the co-owned data sensitivity value ranges in [0,…,1].

$$has[d] = l$$
$$where\ l\ \in \mathbb{R}$$
$$senstivitiy[l] \in [0, \dots, \mathbb{R}]$$

We now present mathematical concepts of the proposed approach. The use of mathematics here helps us to ensure the construction of correct flow control of co-owned data in OSNs since it is precise and unambiguous, unlike natural language. It forces us to think deeply about the system's behaviour, and allows formal analysis.

a) **Definition 1:** <u>*Assigns to*</u>; The developed framework assigns roles to users, sensitivity value to co-owned data, and reputation values to users.

- **reputed** $\in$ USERS $\rightarrow \mathbb{Z}$

It is a *total function* that relates each element of the source with exactly one element of the target. Each user in the system has only one reputation value. None of the users should be assigned more than one reputation value. However, one reputation value can be given to more than one user in the system.

reputed(u,r) means that user *u* is assigned to the reputation value *r*.

$\forall$u.(u $\in$ USERS $\wedge$ r $\in \mathbb{R}$) $\Longrightarrow$ reputed(u,r)

- **has** $\in$ **co-owned** $\rightarrow \mathbb{Z}$

t is a *total function* that relates each element of the source with exactly one element of the target. Each co-owned data in the system has only one data sensitivity value. None of the co-owned data should be assigned to more than one sensitivity value. However, one sensitivity value can be given to more than one co-owned data in the system.

has(d,l) means that co-owned data d is assigned to the sensitivity value l

$\forall d.(d \in$ *co-owned* $\wedge\, l \in [0,\dots,1]) \Longrightarrow$ *has(d,l)*

- **access** $\in$ **targetedgroup**$\leftrightarrow$**co-owned**

Let targeted group be a subset of USERS which involves users who are chosen for being an accessor/viewer for co-owned data. It is the set of relations between users and co-owned data in the system. It means that the users in targeted group set can access to co-owned data.

$\forall$u.(u $\in$ targetedgroup $\wedge$ d $\in$ co-owned$\Longrightarrow$access(u,d)

b) **Definition 2.** *Re-sharing: Each* shared co-owned data, which is held by the targeted group, might be shared with a new group of people or person. Figure 2 illustrates the general structure of co-owned data sharing process and introduces notions and the requirements of the system. In the figure, Re-Share event happens only if co-owned data are accessed by the targeted group. The first condition is on Re-share and it is defined as follows;

$\forall$ d. (d $\in$ co-owned) $\wedge$ $\forall$u. (u $\in$ targetedgroup) $\wedge$ access (u,d) $\Longrightarrow$ Re-share(u,d)$\wedge$ $\forall$u. (u $\in$ USERS)

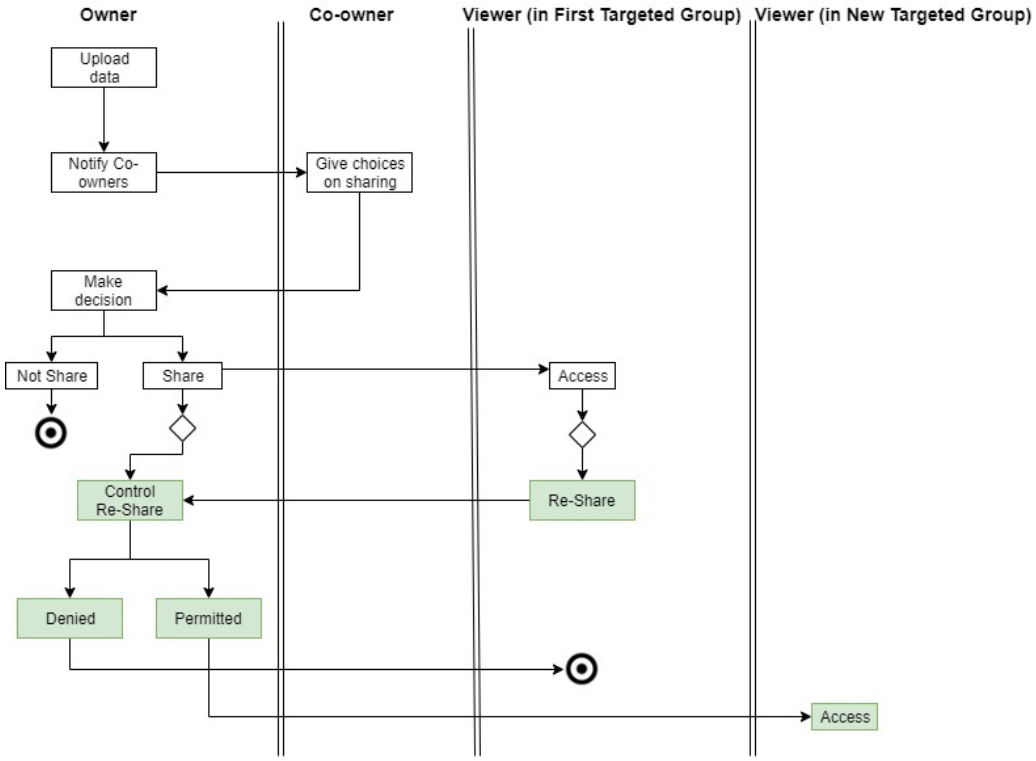

Figure 2: Data Sharing Process Diagram

c) **Definition 3**.*Control Re-sharing and Conditions*: Any shared co-owned data requires re-sharing specifications on controlling or not controlling the flow of co-owned data for the next targeted group. This means that the data owner can either choose to control or not to control the flow of shared co-owned data in OSNs. The control flow is done only if the data owner wants to control the flow of shared co-owned data. With the developed framework, OSN platform needs to control the flow of shared co-owned data. The main purpose here is to ensure that the high sensitive co-owned data is in the circle of trusted people who are not expected to cause any privacy issues with re-sharing the high sensitive co-owned data.

$\forall$ d (access(u,d) $\wedge$ Re-share(u,d)) $\Longrightarrow$ Control Re-share(reputed(u,r), has(d,l))

Flow of co-owned data is controlled when Re-Share happens. On the other hand, Control Re-share(reputed(u,r), has(d,l)) is an activity/ event where users' reputation and co-owned data sensitivity are used as check points. These check points have conditions, which are as follows;

- high co-owned data should not be flown to users whose reputation is not high. This ensures that high sensitive shared co-owned data will never been accessed by users who have leaked users' privacy in the past co-owned data sharing processes. Definition 4 gives the formal modelling and its conditions on co-owned data sensitivity class and users' reputation class. The system will never allow high sensitive data flow to users whose reputation class is not high.

d) **Definition 4.** $\forall d,u.(d,u \in Control\ Re\text{-}share(reputed(u,r),has(d,l)) \wedge \forall d.\ d \in has(d,l) \wedge (value[l] \in high) \wedge \forall u.\ u \in reputed(u,r) \wedge (value[r] \in high) \Longrightarrow access(u,d)$

- Another restriction is on medium sensitive co-owned data (Note: the classes of co-owned data sensitivity are high, medium, and low). Medium sensitive co-owned data might also cause security issues if it is shared with users, whose reputation values are low. Therefore, the system should never allow medium sensitive data flow to low reputed users.

e) **Definition 5.** $\forall$d,u. (d,u $\in$ Control Re-share (reputed(u,r), has(d,l)) $\wedge$ $\forall$d. (d $\in$ has(d,l)$\wedge$(value[l] $\in$ medium)$\wedge$ $\forall$u. (u $\in$ reputed(u,r)$\wedge$(value[r] $\geq$medium) $\Longrightarrow$ access(u,d)

In Definition 4 and Definition 5, conditions are on co-owned data sensitivity and users' reputation values. Definition 4 checks if co-owned data belongs to high sensitive class and users' reputation is high who are targeted for high sensitive co-owned data, then access permission is allowed for those users. Definition5 checks if co-owned data belongs to the medium sensitive class and users' reputation values are at least medium and high, then access permission is allowed for those users.

A. Variables' Normalisation

In the formal modelling and analysis of a system, it is important to know how the system needs to behave under which circumstances. With this respect, this section of the

work specifies the requirements and the functions. The reputation and the co-owned data sensitivity values are the real numbers, however, we use integer numbers in this section. The reason being we use Event B tool in order to prove defined formal models and Event B tool does not provide the real numbers' usage, therefore, we convert real numbers to integers.

The first integer, commonly known as the significant, is to be interpreted as a float with the floating point occurring after the first two decimal digits. The second integer is to be interpreted as the power of 10, commonly known as the base, which is to be multiplied to the significant in order to give the real value of the floating point number (significant x 10^{base}) [21]. In order to do conversion and not missing any values in the system, we multiply the reputation values and the sensitivity value with base two.

Normalisation factor[reputation] = (significant x 10^2) $=\Rightarrow n$ where $n \in [0-Z]$
Normalisation factor[sensitivity] = (significant x 10^2) $=\Rightarrow n$ where $n \in [0-10]$

We give the conversion of real numbers into the integer numbers with the dimensions and the application of normalisation factors for all units in the data sensitivity and the reputation values.

Table 2 explains mapped values of the reputation values and the co-owned data sensitivity values after applying the normalisation on those values.

Table . Values as Reel Numbers

Elements of Xi Set	Definition	Class
reputation	$\forall_r.r \in Z$	--------
sensitivity	$\forall_l.l \in Z$	--------
reputation	$r \mid r \in Z \wedge r \in [0\text{-}130)$	Low
reputation	$r \mid r \in Z \wedge r \in [130\text{-}290)$	Medium
reputation	$r \mid r \in Z \wedge r \in [290\text{-}400]$	High
sensitivity	$l \mid l \in Z \wedge l \in [0\text{-}40)$	Low
sensitivity	$l \mid l \in Z \wedge l \in [40\text{-}70)$	Medium
sensitivity	$l \mid l \in Z \wedge l \in [70\text{-}100]$	High

VI. Implementation with Event-B and the Rodin Platform

Context machine presents the sets, constants, and axioms of the system. USERS, DATA, and permission are the sets that are used in the whole system. Constants define the variables whose values remain same during the system development. In our case, users, targetedgroup, yes, and no are the variables whose values are stable. targetedgroup represents the first targeted group of people for co-owned data and users indicates any user in OSN platform.

CONTEXT
CoownedDataC
SETS
USERS
DATA
Permission
CONSTANTS
Users
Targetedgroup
Yes
No
AXIOMS
axm1: USERS ≠ Ø
axm2: DATA≠ Ø
axm3: users⊆USERS
axm4:targetedgroup⊆USERS
axm5: permission={yes,no}
axm6: yes6=no
END

Machine *CoownedDataM* introduces the abstract machine which uses the sets. The names of variables, whose values comprise the machine, state the machine declared within the *variables* clause. The *invariant* provides information concerning state (i.e. variables) of the machine, including the types of variables and restrictions on their values for the state to be considered meaningful.

CoownedDataM machine represents an OSN platform which uses the developed framework with users' reputation values and co-owned data sensitivity value for controlling shared co-owned data flow. The machine sees *CoownedDataC*, variables are *reputed*, *co-owned*, *has*, and *access*. Details of each invariant are as follows;

- $Reputed \in USERS \Rightarrow Z$

Each user in the members set has a reputation value which is named reputed and it is assigned to a numerical value in Z. A user can only have one reputation value but one reputation value can be given to more than one user.

- $has \in coowned \Rightarrow Z$

Each co-owned data has only one value in Z but one data sensitivity value can be assigned to more than one data

- $coowned \subset DATA$

Each data which is an element of coowned set is also an element of DATA. This is needed because every data in co-owned set needs to have a sensitivity value.

- $Access \in targetedgroup \leftrightarrow coowned$

Each member in the first targeted group set has an access data in co-owned set. This is an interesting invariant because the system's controlling point starts from this invariant. When a user in targeted group has access to co-owned data, the user can re-share the data. However, this work introduces that the system has control points for co-owned data flow.

The abstract machine is responsible for assigning users' reputation, co-owned data sensitivity value, and allows first targeted group for accessing co-owned data. There are three events in the machine, *assignusersreputation(u,r)*, *ssigncoowneddatasensitivity(d,l)*,and

accessfirsttargetedgroupcoowneddata(u,d) respectively. The machine's behaviours on the given events is as follows;

MACHINE
CoownedDataM
SEES
CoownedDataC
VARIABLES
reputed
co-owned
has
access
INVARIANTS
userreputation: reputed∈USERS→ℤ
coowneddata: coowned ⊂DATA
coowneddatasensitivity: has∈coowned→ℤ
accessrelation: access∈targetedgroup↔coowned
EVENTS
assignusersreputation≙
STATUS
ordinary
ANY
u, r
WHERE
grd1: u ∈ USERS
grd2:r ∈ ℤ
THEN
act1: reputed(u) := r
END
assigncoowneddatasensitivity≙
STATUS
ordinary
ANY
d, l
WHERE
grd1: d ∈ DATA
grd2:l ∈ ℤ
grd3: d∉ coowned
THEN
act1: coowned= coowned ∪ {*d*}
has(d)=l
END assignfirsttargetedgrouptocoowneddata≙
STATUS
ordinary
ANY
u, d
WHERE
grd1: u ∈ targeetdgroup
grd2:d ∈ coowned
THEN
act1:access=access ∪ {u↦d}
END
END

Event *assignusersreputation*(*u,r*): The event takes two variables *u,r* as guards, these are necessary conditions for the event to occur. This event picks any user *u* from the USERS set and assigns a reputation value *r* to the user *u, where r ∈ Z.*

Event *assigncoowneddatasensitivity(d,l):* This events takes *d,l* variables as guards. The data d is the member of DATA but not a member in the coowned set. This event adds the *d* to the coowned set and assigns an integer value to data as a value which indicates the sensitivity value for the data.

Event *accessfirsttargetedgroupcoowneddata(u,d):* It is an event that allows access user *u* to data *d.* As it is aforementioned that this is first condition for controlling co-owned flow data because the targeted group's users need to have access and start dissemination of co-owned data.

B. Refinement

Given machine shows what behaviour is required for an implementation. Now, we explain how the given behaviour should be achieved (see CoownedDataMR). Refinement machine includes aspects of how the behaviours are to be achieved in the implementation. The refined machine represents the addition of more detail to the initial abstract machine. The refined machine is now able to control the flow of shared coowned data based on the conditions on shared

coowned data sensitivity values and users' reputation values. The refined machine's invariants have more specified conditions for making sure that the sensitive data does not flow to unwanted members in the system. The refinements on the variables, invariants, and events are as follows;

- *reshare, controlledaccess, reshareddata* and *permitted* (*variables*): Given variables are the new variables in the refined machine. We now explain given variables' detailed definition with related invariants.
- *resharedtargetedgroup*: *reshare* ∈ *targetedgroup* → *coowned*; This invariant introduces total function *reshare* from targetedgroup et to coowned set. Any user in the targeted group can *reshare* co-owned data which was accessed by him. Access has been defined in the abstract machine.
- *reshareddatafromcoowned: reshareddata* ⊆ *coowned*

Any data in reshared dataset has to be an element of coowned set

- *permittedordenied: permitted* ∈ *permission*

*I*t is a new invariant which can have only two values either yes or no, which are constants of permission set in the context machine.

- *Controlledaccessisusertoreshared: controlledaccess* ∈ *reshareddata* → *permission*

It is a checkpoint of re-shared co-owned data which shows whether thepermission is allowed (i.e.yes) or denied (i.e.no).

- *resharingcontrol:* It introduces the condition on the re-shared co-owned data with*;*

∀d.(d ∈ reshareddata) ∧ (∀. (u ∈ users)) ∧ d ∈ ran(access)⇒permitted=yes

Any user u in users set is permitted to any data d in reshareddata set where the data d has to be an element of coowned .set.

- *resharingaccesscontrolpoint1:* It is a refinement on event access first targeted group coowned data in the abstract machine. As it is afore mentioned that all refinements are on event access first targetedgroup coowned data because of the starting point of dissemination of co-owned data. In order to access re-shared co-owned data, the system should go over various guards. The details of each guard's is as follows;
  - u ∈ users: User u in the users' set
  - d ∈ coowned and d ∉ reshareddata: Data has to be accessed by co-owners and then it can be re-shared. Therefore, data d is an element of coowned set but not an element of reshared data set.
  - permitted=no: At the beginning, the data is not permitted for dissemination
  - Conditions are on the data sensitivity and the users' reputation values. Therefore, it is important to check users' reputation values with r ∈ Z ∧ 290 < r ≤400, which ensures that the user u's reputation r is in high class and co-owned data sensitivity value with l ∈ Z ∧ 70 < l ≤ 100is high sensitive (see Definition 4)

When all guards are correct, the event act1 and act2 occur

Reshareddata := reshareddata ∪ d: Data d is moved to reshared data and controlledaccess:= controlledaccess ⊳ ∀ u. (u ∈ users) ∧ reputed (u) := r ∧ ∀d. (d ∈ reshareddata) ∧ has(d) := l ⇒ permitted=yes: All users whose reputation values are in the range of guard (grd7), are permitted to access the re-shared co-owned data which has high sensitivity (grd6)

- resharingaccesscontrolpoint2: It is the second refinement on event *accessfirsttargetedgroupcoowneddata* in the abstract machine. The details of each guard's in the event are as follows;
  - u ∈ users: User u in the users set.
  - d∈coownedandd6∈reshareddata: Data has been to be accessed by co-owners and then it can be re-shared. Therefore, data d is an element of coowned set but not an element of reshareddata set
  - permitted=no At the beginning, the data is not permitted for dissemination
  - Conditions are on the data sensitivity value and the users' reputation values. Therefore, it is important to check users' reputation values with r ∈ Z ∧ 130 < r ≤290, which ensures that the user u' reputation r is in at least medium class and co-owned data sensitivity value with l ∈Z ∧ 40 < l ≤ 70 is medium sensitive (see Definition 5)

When all guards are correct, the event act1 and act2 occur *reshareddata:= reshareddata ∪ d*:

Data d is moved to reshared data and *controlledaccess:= controlledaccess ⊳ ∀u. (u ∈ users)∧reputed(u):=r ∧ ∀d. (d ∈ reshareddata) ∧ has(d) :=l ⇒permitted=yes:*

All users whose reputation values are in the range of guard (grd6), are permitted to access the re-shared co-owned data which has high sensitivity (grd5)

MACHINE

CoownedDataMR

REFINES

CoownedDataM

VARIABLES

reshare

Controlledaccess

reshareddata

permitted

INVARIANTS

resharestargetedgroup: reshare ∈ targetedgroup →Coowned

reshareddatafromCoowned:reshareddata ⊆Coowned

permittedordenied:permitted ∈permission

Controlledaccessisusertoreshared: Controledaccess ∈ reshareddata → permission

resahringcontrol: ∀d. (d∈reshareddata) ∧ ∀u.(u ∈users) ∧ d ∈ ran(access) ⟹ permitted=yes

EVENTS

resharingaccesscontrolpoint1≙

STATUS

ordinary

REFINES

accessfirsttargetedgrouptocoowneddata

ANY

u, d

WHERE

grd1:u ∈ users

grd2: d ∉ reshareddata

grd3: d ∈ coowned

grd4:permitted=no

grd5:l ∈ ℤ ∧ (70<l≤100)

grd6:r ∈ ℤ ∧ (290 < r ≤ 400)

THEN

act1: reshareddata := reshareddata ∪ {d}

act2: Controlledaccess= Controlledaccess ⩤ (∀u. (u ∈ users ∧ reputed(u)=r) ∧ (∀d. (d ∈ reshareddata ∧ has(d)=l) ⟹ permitted=yes)

END

resharingaccesscontrolpoint2≙

STATUS

ordinary

REFINES

accessfirsttargetedgrouptocoowneddata

ANY

u, d

WHERE

grd1:u ∈ users

grd2: d ∉ reshareddata

grd3: d ∈ coowned

grd4:permitted=no

grd5:l ∈ ℤ ∧ (40<l≤70)

grd6:r ∈ ℤ ∧ (130 < r ≤ 290)

THEN

act1: reshareddata := reshareddata ∪ {*d*}

act2: Controlledaccess= Controlledaccess ⩤ (∀u. (u ∈ users ∧ reputed(u)=r) ∧

(∀d. (d ∈ reshareddata ∧ has(d)=l) ⟹ permitted=yes)

END

VII. SUMMARY

In this section, we have presented a formal specification and formal modelling regarding the future flow of shared coowned data in OSNs' platforms. We have first started with the diagram of the proposed work, which covers users' and data interactions with the specifications of activities. The proposed work formal modelling requires definition of the sets, relationships between sets, and roles of each attributes in the system as a second step. It is needed to give the most important part of the system with co-owned data sharing process diagram to highlight the most focused activities. We have shown the needs of highlighting in Figure 2. Green boxes present the control flow of co-owned data in the developed framework. The focused part of Figure 2 is formalised. Using the defined requirements, functions, relations and sets, we have created a machine in Event-B defines the control future flow of co-owned data in the system.

The abstract machine is the first level which does not specify the conditions for re-sharing action in the system. In the refinement machine, we have refinement on invariants and events. The refinement machines define the conditions on co-owned data sensitivity and users' reputations for either allowing the flow of co-owned data or disallowing the flow.

(∀d·(d ∈ reshared ∧ class[sensitivity]> (∀u·(u ∈ members ∧ class[reputation]))) ⇒ (u |→d ∉ access) ((∀d·(d ∈ reshared ∧ class[has(d)]) > (∀u·(u ∈ members ∧ class[reputation(u)])))=⇒(u |→d ∉ access)

Given expression summarises the purpose of the developed framework's control point. It does not allow flow of any element of co-owned data, which has high class sensitivity, to any user in the system, whose reputation has lower class value than co-owned data sensitivity class value. The class values of the reputation and the class values of the co-owned data sensitivity are given on Table 2.

Figure 3 illustrates the structure of re-sharing control in the system. As it is aforementioned that the developed framework checks re-sharing of the shared co-owned data *only if* the owner wants to control future flow of the data. When the first group of users (*the first targeted group*) access the shared co-owned data and intends to share it with the new group of people, the control machine starts fine-grained checking on the co-owned data sensitivity and the users' reputation, who are in the new targeted group, whose members are intended to have permission to access the shared co-owned data. Fine-grained control means that new sharing is individual, not group-based. The data is available to only those users whose reputation' class value is greater than the co-owned data sensitivity's class value.

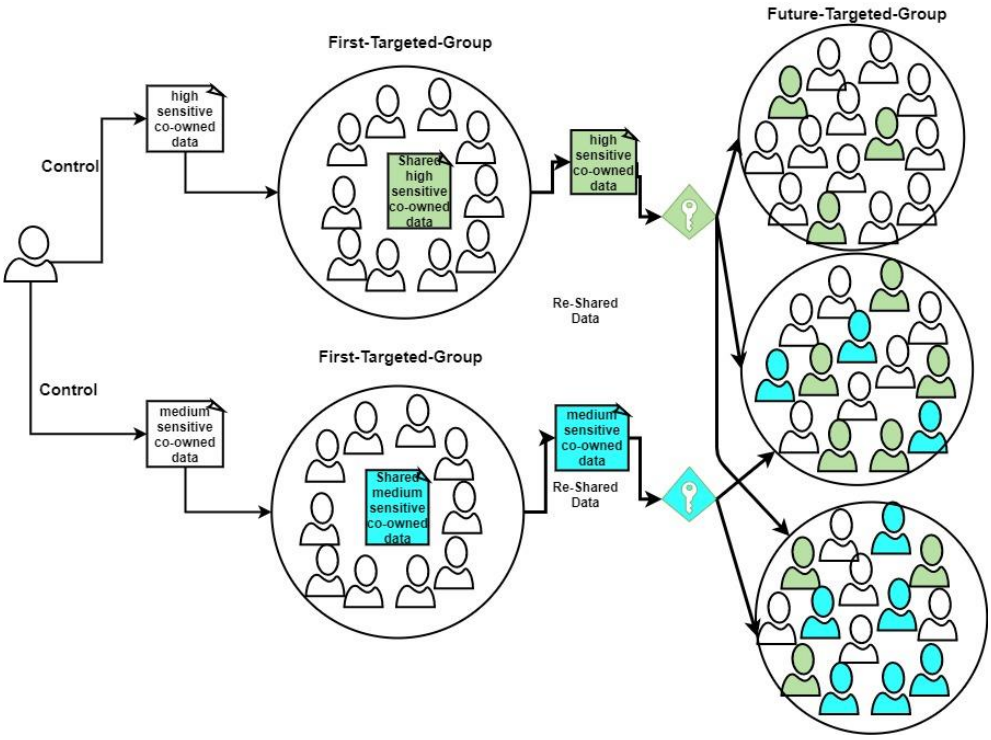

Figure 3: Control Machine Re-Sharing Control Structure

The proposed model and specifications on the machines have been used in Trusty online social network (http://www.trusty.gen.tr/).Trustis an online social networks in which users reputation values and co-owned data sensitivity are used to control flow of shared data. Trusty currently has more than one thousand active users on. When we check the control flow activities on the shared contents in Trusty, we have observed that users want to control specifically the high sensitive contents of co-owned data in OSNs. It has shown that our approach has been suitable not only for implementation in a real-world application but also for controlling flow of sensitive contents.

VIII. CONCLUSION

Formal modelling is an advantage for expressing good properties of specification and proving the obligations in a system. Therefore, this work has presented formal modelling and verification of controlling of shared co-owned data future flow by using Event-B formal modelling language. OSNs' platforms have commonly been used by people, people communicate to each other via data. OSNs' users are let to control the flow of data for the first targeted group, however, they do not have control of flow once the data is on the targeted group's people hands. In this work, we have first shown the problem of the current OSNs on controlling shared contents of co-owned data. Our approach aims to assign the reputation values to users and sensitivity values to co-owned data and use those values for controlling co-owned data flow in OSNs. Formal modelling in Event-B allowed us to completely define and verify the control flow and prove the accuracy of the flow control of shared coowned data. In this work, we use OSNs platforms as a case study, however, the specifications and functions are enough general to cover not only OSNs but also any system that has similar features with the proposed work.

This work contributes to existing knowledge of OSNs' data flows by providing a way of controlling the future flow for shared co-owned data in OSNs. We have shown that how to take an OSN to present the use of Event-B, for not just changing states, but also controlling movement of shared co-owned data. This work has also explained in detail a shared co-owned data control flow to make sure that the high sensitive data never flows to people whose reputation values are not high.

REFERENCES

[1] *Akkuzu, G., Aziz, B., & Adda, M. (2019, January). Fuzzy logic decisionbased collaborative privacy management framework for online social net-works. In 3rd International Workshop on FORmal Methods for SecurityEngineering: ForSE.*

[2] Facebook. 2020. website: https://www.ballantine.com/facebook-algorithm-changes/:text=Facebook

[3] Akkuzu, G., Aziz, B., & Adda, M. (2019, October). Advantages of havingusers' trust and reputation values on data sharing process in online socialnetworks. In 2019 Sixth International Conference on Social Networks Anal-ysis, Management and Security (SNAMS) (pp. 189-195). IEEE.

[4] *Akkuzu, G., Aziz, B., & Adda, M. (2019, October). Advantages of havingusers' trust and reputation values on data sharing process in online socialnetworks. In 2019 Sixth International Conference on Social Networks Anal-ysis, Management and Security (SNAMS) (pp. 189-195). IEEE.*

[5] Akkuzu, G., Aziz, B., & Adda, M. (2020). Towards consensus-based groupdecision making for co-owned data sharing in online social networks. IEEEAccess, 8, 91311-91325.

[6] Monin, J.F. (2003), Understanding Formal Methods, Springer, 2003, XVI,276.

[7] Joseph, N. S. (2014). Collaborative data sharing in online social networkresolving privacy risk and sharing loss. IOSR-JCE) eISSN, 2278-0661

[8] Abrial, J. R. (2010). Modeling in Event-B: system and software engineering.Cambridge University Press.

[9] Marin, A., & Wellman, B. (2011). Social network analysis: An introduction.The SAGE handbook of social network analysis, 11.

[10] Hanneman, R. A., & Riddle, M. (2005). Introduction to social networkmethods.

[11] Ali, B., Villegas, W., & Maheswaran, M. (2007, October). A trust basedapproach for protecting user data in social networks. In Proceedings of the2007 conference of the center for advanced studies on Collaborative research(pp. 288-293).

[12] Fong, P. W., & Siahaan, I. (2011, June). Relationship-based access con-trol policies and their policy languages. In Proceedings of the 16th ACMsymposium on Access control models and technologies (pp. 51-60).

[13] Liben-Nowell, D., & Kleinberg, J. (2008). Tracing information flow on aglobal scale using Internet chain-letter data. Proceedings of the nationalacademy of sciences, 105(12), 4633-4638.

[14] Lu, Y., Wang, W., Bhargava, B., & Xu, D. (2006). Trust-based privacy preservation for peer-to-peer data sharing. IEEE Transactions on Systems,Man, and Cybernetics-Part A: Systems and Humans, 36(3), 498-502.

[15] Krishnan, R., Sandhu, R., & Ranganathan, K. (2007, June). PEI modelstowards scalable, usable and high-assurance information sharing. In Proceed-ings of the 12th ACM symposium on Access control models and technologies(pp. 145-150).

[16] Zdancewic, S., & Myers, A. C. (2001, April). Secure information flowand CPS. In European Symposium on Programming (pp. 46-61). Springer,Berlin, Heidelberg.

[17] Baden, R., Bender, A., Spring, N., Bhattacharjee, B., & Starin, D. (2009,August). Persona: an online social network with user-defined privacy. In Pro-ceedings of the ACM SIGCOMM 2009 conference on Data communication(pp. 135-146).

[18] Abrial, J. R., M´etayer, C., & Voisin, L. (2005). Event-B language. RodinDeliverable, 3.

[19] Jiang, W., Wu, J., Li, F., Wang, G., & Zheng, H. (2015). Trust evaluationin online social networks using generalized network flow. IEEE Transactionson Computers, 65(3), 952-963.

[20] Bhargava, B., Angin, P., Ranchal, R., Sivakumar, R., Sinclair, A., & Lin-derman, M. (2012). A Trust-based Approach for Secure Data Disseminationin a Mobile Peer-to-Peer Network of AVs. International Journal of Next-generation Computing, 3(1).

[21] Gibson, J. P., & M´ery, D. (2018). Explicit modelling of physical measures:from Event-B to Java. arXiv preprint arXiv:1805.05517.

[22] Vishwamitra, N., Li, Y., Wang, K., Hu, H., Caine, K., & Ahn, G. J.(2017, June). Towards pii-based multiparty access control for photo sharingin online social networks. In Proceedings of the 22nd ACM on Symposiumon Access Control Models and Technologies (pp. 155-166).

[23] Amato, F., Moscato, F., Moscato, V., & Colace, F. (2018). Improvingsecurity in cloud by formal modeling of IaaS resources. Future GenerationComputer Systems, 87, 754-764.

[24] Souri, A., & Norouzi, M. (2019). A state-of-the-art survey on formal ver-ification of the internet of things applications. Journal of Service ScienceResearch, 11(1), 47-67.

[25] Luckcuck, M., Farrell, M., Dennis, L. A., Dixon, C., & Fisher, M. (2019).Formal specification and verification of autonomous robotic systems: A sur-vey. ACM Computing Surveys (CSUR), 52(5), 1-41

[26] Shukla, N., Pandey, M., & Srivastava, S. (2019). Formal modeling andverification of software-defined networks: A survey. International Journal ofNetwork Management, 29(5), e2082.

[27] Kamel, O., Chaoui, A., & Gharzouli, M. (2017, December). Towards aFormal Modeling of Cloud Services during the Life-cycle of Service LevelAgreement. In Proceedings of the International Conference on Big Data andInternet of Thing (pp. 115-119).

[28] Alvim, M. S., Knight, S., & Valencia, F. (2019). Toward a Formal Modelfor Group Polarization in Social Networks. In The Art of Modelling Compu-tational Systems: A Journey from Logic and Concurrency to Security andPrivacy (pp. 419-441). Springer, Cham.

[29] Hansen, J. U. (2019). Reasoning about opinion dynamics in social networks.Journal of Logic and Computation, 29(7), 1121-1137.

[30] Abdulrahman, R., Holton, D. R. W., Neagu, D., & Ridley, M. (2011, June).Formal specification of multi agent system for historical information retrievalfrom online social networks. In KES International Symposium on Agent andMulti-Agent Systems: Technologies and Applications (pp. 84-93). Springer,Berlin, Heidelberg