Twitlang(er)
– Interactions Modeling Language (and Interpreter) for Twitter –

Francesco Tiezzi
School of Science and Technology
Computer Science Division
University of Camerino

CINA: Final General Meeting
Civitanova Marche (Italy) - January 21st, 2016

Joint work with Rocco De Nicola (IMT), Alessandro Maggi (IMT), Marinella Petrocchi (CNR), Angelo Spognardi (DTU)
Twitter

What Twitter is:
- Socially-centric platform: microblogging site, social network, information network, ...
- Service provider that enables users to send and read 140-character messages (tweets)

Twitter’s success:
- Launched in 2006
- Stats (August 2014): 271 million of (active) users, with an average of 500 million of tweets per day
- Popular users: politicians, actors, singers, sportsmen, mass media, companies (for business promotion), civil protection departments, humanitarian driving forces, ...
Twitter: user account

University of York
@UniOfYork

The University of York is a leading UK university and part of the Russell Group. Follow us for our latest news and views.
York, UK
york.ac.uk
Joined March 2009

Tweets

University of York retweeted
The York Union @YorkUnion - Aug 7
We would like to highlight that our YouTube is now fully updated with all of the events from last year’s programme. youtube.com/channel/UCoUPM...

University of York @UniOfYork - Aug 14
"The sheer range of opportunities on offer at York is phenomenal" - meet Ben on.fb.me/1J88uwq #humansofyork

Who to follow
Servizio Pubblico @Serv...
Carlo Pinciroli @Ipinoy
Dmytro Karameshuk @kara...

Find trends
#PerAmoreDelloSport
Promoted by Sky Sport
#Bangkok
#5SOSFAMisProudOf5SOS
#earthquake
#mondaymotivation
Bieber
David Milliband
Twitter users can build and manage **social relationships**, according to which messages are delivered, by means of **actions** [Follow] and [Unfollow].

Types of relationships between two users $A$ and $B$:

- $A$ **follows** $B$: the tweets posted by $B$ appear on $A$’s timeline
- $B$ **follows** $A$
- $A$ and $B$ **follow each other**
- **No relationship** between $A$ and $B$
Twitter: user actions

Tweet a new 140-character message, including mentions, hashtags, urls, pictures

Retweet a message to spread to followers what the user thinks particularly worth of notice

Reply to a message posted by a user

Delete a tweet

Undo a retweet
Motivation of the work: What Happens to My Tweets?

Focus
We focus on one of the core aspects of Twitter, which makes it so popular and widespread: its interaction model.
Motivation of the work: What Happens to My Tweets?

Focus
We focus on one of the core aspects of Twitter, which makes it so popular and widespread: its interaction model.

Motivations
- Any Twitter user should precisely know the dynamics of his tweets:
  - which are the accounts directly reachable by my tweets?
  - what happens if I delete one of my tweets? . . .
- A conscious usage of Twitter is even more crucial when it is used as a communication media to support (critical) collaborative work.
Motivation of the work: What Happens to My Tweets?

Focus
We focus on one of the core aspects of Twitter, which makes it so popular and widespread: its interaction model.

Motivations
- Any Twitter user should **precisely know the dynamics of his tweets**
  - which are the accounts directly reachable by my tweets?
  - what happens if I delete one of my tweets? . . .
- A conscious usage of Twitter is even more crucial when it is used as a communication media to support (critical) **collaborative work**

Challenge
Despite the apparent simplicity of Twitter interactions, the achievement of a **full user experience-awareness** should not be given for granted.
Motivation of the work: What Happens to My Tweets?

Focus
We focus on one of the core aspects of Twitter, which makes it so popular and widespread: its interaction model

Motivations
- Any Twitter user should **precisely know the dynamics of his tweets**
  - which are the accounts directly reachable by my tweets?
  - what happens if I delete one of my tweets? . . .
- A conscious usage of Twitter is even more crucial when it is used as a communication media to support (critical) **collaborative work**

Challenge
Despite the apparent simplicity of Twitter interactions, the achievement of a **full user experience-awareness** should not be given for granted

Long term goal
Provide analysis techniques to distinguish genuine Twitter accounts from **anomalous** ones (spammers, fake followers, . . .)
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet

@mickey follows @goofy

@mickey

@goofy
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet

@mickey follows @goofy

F. Tiezzi (UNICAM)
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet

@mickey follows @goofy

@mickey sent a notification to @goofy.
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet

@mickey follows @goofy

@mickey

@goofy
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet

@mickey

@goofy

follows

---

F. Tiezzi (UNICAM)
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet

@mickey follows @goofy follows @donald
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet

@mickey follows @goofy follows @donald
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet

@mickey follows @goofy follows @donald
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet

@mickey follows @goofy follows @donald
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet

Even these simple examples have effects that could be not fully intuitive for the community
Examples of Twitter interactions

The effects of the removal actions in these interactions are quite different:

1. tweet — reply — delete the tweet
2. tweet — retweet — undo the retweet
3. tweet — retweet — retweet the retweet — delete the tweet

Even these simple examples have effects that could be not fully intuitive for the community.

Other examples lead to more counterintuitive effects.
Example with counterintuitive effects

@mickey follows @donald

@goofy
Example with counterintuitive effects

@mickey follows @donald

@donald great work by @goofy on #formalmethods and Twitter! Let’s start a collaboration!
Example with counterintuitive effects

@mickey follows @donald

@goofy
Example with counterintuitive effects

@mickey  doesn’t go for it, waste of time

@mickey  follows  @donald

@donald

@goofy
Example with counterintuitive effects

@mickey follows @donald

@goofy

expand

F. Tiezzi (UNICAM) Twitlang(er) January 21st, 2016
Example with counterintuitive effects

@mickey follows @donald
@goofy

expand
Example with counterintuitive effects

Notifications
All / People you follow

mickey @mickey · Jun 11
@donald great work by @goofy on #formalmethods and Twitter! Let's start a collaboration!

10:45 AM - 11 Jun 2014 · Details
Collapse

Reply to @mickey

donald @donald · Jun 11
@mickey don't go for it, waste of time
Expand

Go to Reply · Retweet · Favorite · More
Example with counterintuitive effects
Example with counterintuitive effects

@mickey follows @donald

@goofy
Our proposal

The issue

Fully conscious usage of Twitter calls for a **precise awareness** of the **dynamics regulating message spreading**
Our proposal

The issue
Fully conscious usage of Twitter calls for a **precise awareness** of the **dynamics regulating message spreading**

Twitlang
formal **language** to **rigorously model** interactions among Twitter accounts

Twitlanger
Maude implementation of Twitlang, enabling automatic verification of communication properties of Twitter accounts via **model checking**
Twitlang
Twitlang syntax: account

\[ u : T : N : F : B \]
Twitlang syntax: account

username (e.g., @UniOfYork, @micheky,...)

- timeline
- notification list
- followings
- behaviour

F. Tiezzi (UNICAM)
Twitlang syntax: account

\[ u : T : N : F : B \]

- **username** \( u \) uniquely identifies the account
- **timeline** \( T \) is the list of messages received from the account’s followings or sent by the account
- **notifications list** \( N \) contains the messages where the account’s username is mentioned and the replies to account’s messages
- **followings list** \( F \) contains the usernames of following accounts
- **behaviour** \( B \) is a process performing Twitter actions
Twitlang syntax: messages

A message $m$ is a data tuple of the form

$$\langle id_{cur}, id_{ret}, id_{rep}, text, u_a, u_l, u_s \rangle$$

where

- $id_{cur}$: id of the (current) message
- $id_{ret}$: id of the original tweet the current message is a retweet of
- $id_{rep}$: id of the message the current message is a reply to
- $text$: textual content of the message
- $u_a$: username of author of the retweeted/replied original message
- $u_l$: username of sender of the last retweet in a retweet chain
- $u_s$: username of sender of the current message
Twitlang syntax: behaviours and actions

Account **behaviours** are modelled by terms of a simple process algebra:

\[
B ::= \text{nil} \mid a.B \mid B_1 + B_2 \mid B_1 | B_2 \mid K \ (\text{with } K \triangleq B)
\]
Twitlang syntax: behaviours and actions

Account behaviours are modelled by terms of a simple process algebra:

\[ B ::= \text{nil} \mid a.B \mid B_1 + B_2 \mid B_1 \parallel B_2 \mid K \ (\text{with } K \triangleq B) \]

Actions \( a \) are the ones provided by Twitter:

- **tweet**: \text{tweet}(\text{text}, x)
- **delete**: \text{delete}(x)
- **find**: \text{find}(P, z)@t
- **retweet**: \text{retweet}(z, y)
- **undo**: \text{undo}(y)
- **reply**: \text{reply}(z, \text{text}, U, x)
- **follow**: \text{follow}(u)
- **unfollow**: \text{unfollow}(u)
Twitlang syntax: networks

A Twitlang specification is a **network** of accounts:

\[
\mathcal{N} ::= u : T : N : F : B \mid \mathcal{N}_1 \parallel \mathcal{N}_2
\]
Twitlang syntax in one slide

(Networks) \( \mathcal{N} ::= u : T : N : F : B \mid \mathcal{N}_1 \parallel \mathcal{N}_2 \)

(Timelines) \( T ::= \epsilon \mid m \mid T_1, T_2 \)

(Notifications) \( N ::= \epsilon \mid m \mid N_1, N_2 \)

(Messages) \( m ::= \langle id_{cur}, id_{ret}, id_{rep}, text, u_a, u_l, u_s \rangle \)

(Followings) \( F ::= \epsilon \mid u \mid F_1, F_2 \)

(Behaviours) \( B ::= \text{nil} \mid a.B \mid B_1 + B_2 \mid B_1 \mid B_2 \mid K \)

(Actions) \( a ::= \text{tweet}(text, x) \mid \text{delete}(x) \)
\( \mid \text{find}(P, z)@t \mid \text{retweet}(z, y) \mid \text{undo}(y) \)
\( \mid \text{reply}(z, text, U, x) \mid \text{follow}(u) \mid \text{unfollow}(u) \)

(Targets) \( t ::= u \mid \text{all} \)
Twitlang syntax: a simple example

Consider again the interaction

- tweet — retweet — retweet the retweet — delete the tweet
Twitlang syntax: a simple example

Consider again the interaction

- tweet — retweet — retweet the retweet — delete the tweet

The initial network is:

\[ @\text{mickey}: \epsilon: \epsilon: \epsilon: B_m \parallel @\text{goofy}: \epsilon: \epsilon: @\text{mickey}: B_g \parallel @\text{donald}: \epsilon: \epsilon: @\text{goofy}: B_d \]

with

\[
\begin{align*}
B_m & = \text{tweet}(Hello, x). \text{find}(\downarrow_7 = @\text{donald}, z)@\text{mickey}. \text{delete}(x). \text{nil} \\
B_g & = \text{find}(\downarrow_7 = @\text{mickey}, z')@\text{goofy}. \text{retweet}(z', y). \text{nil} \\
B_d & = \text{find}(\downarrow_7 = @\text{goofy}, z'')@\text{donald}. \text{retweet}(z'', y'). \text{nil}
\end{align*}
\]

(predicate \(\downarrow_7 = u\) is verified by a message \(m\) if its sender is the username \(u\))
Twitlang syntax: a simple example

Consider again the interaction

- tweet — retweet — retweet the retweet — delete the tweet

The initial network is:

\[ @\text{mickey} : \epsilon : \epsilon : \epsilon : B_m \parallel @\text{goofy} : \epsilon : \epsilon : @\text{mickey} : B_g \parallel @\text{donald} : \epsilon : \epsilon : @\text{goofy} : B_d \]

with

\begin{align*}
B_m &= \text{tweet}(Hello, x). \text{find}(\downarrow_7 = @\text{donald}, z)@\text{mickey}. \text{delete}(x). \text{nil} \\
B_g &= \text{find}(\downarrow_7 = @\text{mickey}, z')@\text{goofy}. \text{retweet}(z', y). \text{nil} \\
B_d &= \text{find}(\downarrow_7 = @\text{goofy}, z'')@\text{donald}. \text{retweet}(z'', y'). \text{nil}
\end{align*}

(predicate \( \downarrow_7 = u \) is verified by a message \( m \) if its sender is the username \( u \))

The evolution of the network is defined by

the operational semantics of Twitlang
Twitlang semantics: a glimpse

Operational semantics

Given in terms of a labeled transition relation on networks $\mathcal{N} \xrightarrow{\lambda} \mathcal{N}'$
Twitlang semantics: a glimpse

Operational semantics

Given in terms of a labeled transition relation on networks $\mathcal{N} \xrightarrow{\lambda} \mathcal{N}'$

Definition of $\xrightarrow{-}$ relies on an auxiliary relation on behaviors $B \xrightarrow{\alpha} B'$
Twitlang semantics: a glimpse

### Operational semantics

Given in terms of a labeled transition relation on networks $N \xrightarrow{\lambda} N'$

Definition of $\mapsto$ relies on an auxiliary relation on behaviors $B \xrightarrow{\alpha} B'$

1. $\text{tweet}(\text{text}, x).B \xrightarrow{\text{tweet}(\text{text}, \text{id})} B[\text{id}/x]$  
   \[\text{retweet}(m, y).B \xrightarrow{\text{retweet}(m, \text{id})} B[\text{id}/y]\]

2. $\text{reply}(m, \text{text}, U, x).B \xrightarrow{\text{reply}(m, (m\downarrow_7 \cdot m\downarrow_5 \cdot \text{mentions}(m\downarrow_4)) \setminus U \cdot \text{text}, \text{id})} B[\text{id}/x]$  
   \[B_1 \xrightarrow{\alpha} B'_1 \quad B_1 \xrightarrow{\alpha} B'_1 \quad B_1 \mid B_2 \xrightarrow{\alpha} B'_1 \mid B_2 \quad \ldots \]
Twitlang semantics: a glimpse

Operational semantics

Given in terms of a labeled transition relation on networks $\mathcal{N} \xrightarrow{\lambda} \mathcal{N}'$.
Twitlang semantics: a glimpse

Operational semantics

Given in terms of a labeled transition relation on networks $\mathcal{N} \xrightarrow{\lambda} \mathcal{N}'$

$$
\begin{align*}
\text{B} & \xrightarrow{\text{tweet(} \text{text, } \text{id} \text{)}} \text{B}' & \text{id} \not\in \text{ids}(T, N, B) \\
\end{align*}
$$

$$
\begin{align*}
\text{u} : T : N : F : B \xrightarrow{\langle \text{id, } \_, \text{, text, } \_, \_, \_, \text{u} \rangle} \text{u} : (T, \langle \text{id, } \_, \text{, text, } \_, \_, \_, \text{u} \rangle) : N : F : B' \\
\end{align*}
$$

$$
\begin{align*}
\text{B} & \xrightarrow{\text{retweet(} \text{m, } \text{id} \text{)}} \text{B}' & \text{id} \not\in \text{ids}(T, N, B) & m \downarrow 7 \neq u \\
\end{align*}
$$

$$
\begin{align*}
\text{u} : T : N : F : B \xrightarrow{\langle \text{id, m\downarrow 2/1, } \_, \text{m\downarrow 4, author(} \text{m} \text{)}, \text{m\downarrow 7, } \text{u} \rangle} \text{u} : (T, \langle \text{id, m\downarrow 2/1, } \_, \text{m\downarrow 4, author(} \text{m} \text{)}, \text{m\downarrow 7, } \text{u} \rangle) : N : F : B' \\
\end{align*}
$$

$$
\begin{align*}
\text{B} & \xrightarrow{\text{reply(} \text{m, text, } \text{id} \text{)}} \text{B}' & \text{id} \not\in \text{ids}(T, N, B) \\
\end{align*}
$$

$$
\begin{align*}
\text{u} : T : N : F : B \xrightarrow{\langle \text{id, } \_, \text{m\downarrow 1, text, m\downarrow 7, } \_, \_, \text{u} \rangle} \text{u} : (T, \langle \text{id, } \_, \text{m\downarrow 1, text, m\downarrow 7, } \_, \_, \text{u} \rangle) : N : F : B' \\
\end{align*}
$$

$$
\begin{align*}
\mathcal{N} \xrightarrow{m} \mathcal{N}' & m \downarrow 1 \not\in \text{ids}(T, N, B) \\
\mathcal{N} \parallel \text{u} : T : N : F : B \xrightarrow{m} \mathcal{N}' \parallel \text{u} : (T \oplus^m F) : (N \oplus^" m) : F : B \\
\end{align*}
$$
Twitlang semantics: a simple example

@mickey : ϵ : ϵ : ϵ : B_m || @goofy : ϵ : ϵ : @mickey : B_g || @donald : ϵ : ϵ : @goofy : B_d
Twitlang semantics: a simple example

@mickey : ε : ε : ε : B_m || @goofy : ε : ε : @mickey : B_g || @donald : ε : ε : @goofy : B_d

\[ m_1 = \langle id_1, _, _, Hello, _, _, @mickey \rangle \]

@mickey : m_1 : ε : ε : B'_m || @goofy : m_1 : ε : @mickey : B_g || @donald : ε : ε : @goofy : B_d
Twitlang semantics: a simple example

@mickey: \( \epsilon : \epsilon : \epsilon : B_m \) || @goofy: \( \epsilon : \epsilon : @mickey : B_g \) || @donald: \( \epsilon : \epsilon : @goofy : B_d \)

\[ m_1 = \langle \text{id}_1, \_ , \_ , \text{Hello} , \_ , \_ , @mickey \rangle \]

@mickey: \( m_1 : \epsilon : \epsilon : B'_m \) || @goofy: \( m_1 : \epsilon : @mickey : B_g \) || @donald: \( \epsilon : \epsilon : @goofy : B_d \)

\[ @goofy:\text{found}(m_1) \]

\[ m_2 = \langle \text{id}_2 , \text{id}_1 , \_ , \text{Hello} , @mickey , @mickey , @goofy \rangle \]

\[ @donald:\text{found}(m_2) \]

\[ m_3 = \langle \text{id}_3 , \text{id}_1 , \_ , \text{Hello} , @mickey , @goofy , @donald \rangle \]

@mickey: \( m_1 : (m_2, m_3) : \epsilon : B'_m \) || @goofy: \( (m_1, m_2) : m_3 : @mickey : \text{nil} \) || @donald: \( (m_2, m_3) : \epsilon : @goofy : \text{nil} \)
Twitlang semantics: a simple example

@mickey : ε : ε : ε : B_m || @goofy : ε : ε : @mickey : B_g || @donald : ε : ε : @goofy : B_d

\( m_1 = \langle id_1, _, _, Hello, _, _, @mickey \rangle \)

@mickey : m_1 : ε : ε : B'_m || @goofy : m_1 : ε : @mickey : B_g || @donald : ε : ε : @goofy : B_d

\( @goofy : \text{found}(m_1) \)        \( m_2 = \langle id_2, id_1, _, Hello, @mickey, @mickey, @goofy \rangle \)

\( @donald : \text{found}(m_2) \)        \( m_3 = \langle id_3, id_1, _, Hello, @mickey, @goofy, @donald \rangle \)

@mickey : m_1 : (m_2, m_3) : ε : B'_m || @goofy : (m_1, m_2) : m_3 : @mickey : nil || @donald : (m_2, m_3) : ε : @goofy : nil

\( @mickey : \text{found}(m_3) \)        \( \text{delete}(id_1) \)

@mickey : ε : ε : ε : nil || @goofy : ε : ε : @mickey : nil || @donald : ε : ε : @goofy : nil
Twitlanger
To enable automatic analysis of Twitlang specifications, we developed **Twitlanger**

**Twitlanger**

Interpreter for Twitlang written in Maude

- it takes a Twitlang term as an input and, automatically or interactively, **explores the computations** arising from the term

- operational rules of the semantics are rendered in terms of **rewrite rules**, plus additional operators and equations
  
  (see Verdejo and Martí-Oliet state-of-the-art Maude implementation of CCS)

- it enables **verification** of communication properties by using the **LTL model checking** facilities offered by the Maude toolset

---

```
Simple example as Twitlanger code

eq um = Mickey : empty : empty : none : (tweet('Hello,x).(search( predP7(Donald),z)@Mickey).delete(x).nil) .

eq ug = Goofy : empty : empty : Mickey : ((search( predP7(Mickey),z')@Goofy).retweet(z',y) .

eq ud = Donald : empty : empty : Goofy : ((search( predP7(Goofy),z'')@Donald).retweet(z'',y').nil) .

eq simpleExample = !( um || ug || ud ) .
```
Twitlanger

To enable automatic analysis of Twitlang specifications, we developed

**Twitlanger**

Interpreter for Twitlang written in Maude

- it takes a Twitlang term as an input and, automatically or interactively, explores the computations arising from the term

- operational rules of the semantics are rendered in terms of rewrite rules, plus additional operators and equations

  (see Verdejo and Martí-Oliet state-of-the-art Maude implementation of CCS)

- it enables verification of communication properties by using the LTL model checking facilities offered by the Maude toolset

**Simple example as Twitlanger code**

```plaintext
eq um = Mickey : empty : empty : none : (tweet('Hello',x).(search( predP7(Donald),z)@Mickey).delete(x).nil) .
eq ug = Goofy : empty : empty : Mickey : ((search( predP7(Mickey),z')@Goofy).retweet(z',y) .
eq ud = Donald : empty : empty : Goofy : ((search( predP7(Goofy),z'')@Donald).retweet(z'',y').nil) .
eq simpleExample = !( um || ug || ud ) .
```

F. Tiezzi (UNICAM)
Twitlanger at work on a scenario from the academic domain

Twitter

- public initiatives -> all (Twitter)
  - students comm -> @Student
  - prof comm -> @Professor
  - internal activities -> @Student @Professor

- external comm -> @University all (Twitter)
- internal comm -> @Office @Professor @Student

- didactic updates -> @Student

F. Tiezzi (UNICAM)
Twitlanger at work on a scenario from the academic domain

Some verified communication properties:

let University send tweets about Academic Year Inauguration and Open Call for Researchers; are they received by Twitter cloud?

YES

let’s suppose that Director needs to issue an internal communication directed at the faculty; is it received by Twitter cloud?

NO, despite its indirect following connection to Director through University

let’s suppose that Professor has to send a message about an exam; is it received by all Students?

NO, students may not follow a professor

Possible solutions:

▶ Professor mentions all students in the tweet; not scalable
▶ Each student interested in the exam must follow the professor account; student accounts are not under the control of the university
▶ Assuming that each student follows Office, the latter could retweet tweets sent by Professor having hashtag ♯exam; students will receive the tweet about the exam, independently from their interest in it

F. Tiezzi (UNICAM)
Twitlanger at work on a scenario from the academic domain

Some verified communication properties:

- let @University send tweets about Academic Year Inauguration and Open Call for Researchers; are they received by Twitter cloud?

YES

- let’s suppose that @Director needs to issue an internal communication directed at the faculty; is it received by Twitter cloud?

NO, despite its indirect following connection to @University

- let’s suppose that @Professor has to send a message about an exam; is it received by all @Student?

NO, students may not follow a professor

Possible solutions:

▶ Professor mentions all students in the tweet; not scalable

▶ Each student interested in the exam must follow the professor account; student accounts are not under the control of the university

▶ Assuming that each student follows @Office, the latter could retweet tweets sent by @Professor having hashtag ♯exam; students will receive the tweet about the exam, independently from their interest in it
Twitlanger at work on a scenario from the academic domain

Some verified communication properties:

- let @University send tweets about Academic Year Inauguration and Open Call for Researchers; are they received by Twitter cloud? **YES**
Twitlanger at work on a scenario from the academic domain

Some verified communication properties:

- let @University send tweets about Academic Year Inauguration and Open Call for Researchers; are they received by Twitter cloud? **YES**

- let’s suppose that @Director needs to issue an internal communication directed at the faculty; is it received by Twitter cloud?

Possible solutions:

- Professor mentions all students in the tweet; not scalable
- Each student interested in the exam must follow the professor account; student accounts are not under the control of the university
- Assuming that each student follows @Office, the latter could retweet tweets sent by @Professor having hashtag Ÿexam; students will receive the tweet about the exam, independently from their interest in it
Twitlanger at work on a scenario from the academic domain

Some verified communication properties:

- let @University send tweets about Academic Year Inauguration and Open Call for Researchers; are they received by Twitter cloud? **YES**
- let’s suppose that @Director needs to issue an *internal* communication directed at the faculty; is it received by Twitter cloud? **NO**, despite its indirect following connection to @Director through @University

Possible solutions:

- Professor mentions all students in the tweet; not scalable
- Each student interested in the exam must follow the professor account; student accounts are not under the control of the university
- Assuming that each student follows @Office, the latter could retweet tweets sent by @Professor having hashtag ♯exam; students will receive the tweet about the exam, independently from their interest in it
Twitlanger at work on a scenario from the academic domain

Some verified communication properties:

- let @University send tweets about Academic Year Inauguration and Open Call for Researchers; are they received by Twitter cloud? **YES**
- let’s suppose that @Director needs to issue an **internal** communication directed at the faculty; is it received by Twitter cloud? **NO**, despite its indirect following connection to @Director through @University
- let’s suppose that @Professor has to send a message about an exam; is it received by all @Student\(_i\)?
Twitlanger at work on a scenario from the academic domain

Some verified communication properties:

- let @University send tweets about Academic Year Inauguration and Open Call for Researchers; are they received by Twitter cloud? YES

- let’s suppose that @Director needs to issue an **internal** communication directed at the faculty; is it received by Twitter cloud? NO, despite its indirect following connection to @Director through @University

- let’s suppose that @Professor has to send a message about an exam; is it received by all @Student? NO, students may not follow a professor
Twitlanger at work on a scenario from the academic domain

Some verified communication properties:

- let @University send tweets about Academic Year Inauguration and Open Call for Researchers; are they received by Twitter cloud? **YES**

- let’s suppose that @Director needs to issue an **internal** communication directed at the faculty; is it received by Twitter cloud? **NO**, despite its indirect following connection to @Director through @University

- let’s suppose that @Professor has to send a message about an exam; is it received by all @Student\_i? **NO**, students may not follow a professor

Possible solutions:

- Professor mentions all students in the tweet; not scalable
- Each student interested in the exam must follow the professor account; student accounts are not under the control of the university
- Assuming that each student follows @Office, the latter could retweet tweets sent by @Professor having hashtag ♯exam; students will receive the tweet about the exam, independently from their interest in it
Twitlanger at work on a scenario from the academic domain

Some verified communication properties:

- let @University send tweets about Academic Year Inauguration and Open Call for Researchers; are they received by Twitter cloud? **YES**
- let’s suppose that @Director needs to issue an **internal** communication directed at the faculty; is it received by Twitter cloud? **NO**, despite its indirect following connection to @Director through @University
- let’s suppose that @Professor has to send a message about an exam; is it received by all @Student? **NO**, students may not follow a professor

Possible solutions:
- Professor mentions all students in the tweet; not scalable
Twitlanger at work on a scenario from the academic domain

Some verified communication properties:

- let @University send tweets about Academic Year Inauguration and Open Call for Researchers; are they received by Twitter cloud? YES

- let’s suppose that @Director needs to issue an internal communication directed at the faculty; is it received by Twitter cloud? NO, despite its indirect following connection to @Director through @University

- let’s suppose that @Professor has to send a message about an exam; is it received by all @Student? NO, students may not follow a professor

Possible solutions:

- Professor mentions all students in the tweet; not scalable
- Each student interested in the exam must follow the professor account; student accounts are not under the control of the university
Twitlanger at work on a scenario from the academic domain

Some verified communication properties:

- let @University send tweets about Academic Year Inauguration and Open Call for Researchers; are they received by Twitter cloud? **YES**

- let’s suppose that @Director needs to issue an **internal** communication directed at the faculty; is it received by Twitter cloud? **NO**, despite its indirect following connection to @Director through @University

- let’s suppose that @Professor has to send a message about an exam; is it received by all @Student? **NO**, students may not follow a professor

Possible solutions:

- Professor mentions all students in the tweet; not scalable
- Each student interested in the exam must follow the professor account; student accounts are not under the control of the university
- Assuming that each student follows @Office, the latter could retweet tweets sent by @Professor having hashtag #exam; students will receive the tweet about the exam, independently from their interest in it
Concluding remarks
Ongoing and future work

- Realising a user-friendly on-line service based on Twitlanger for users not acquainted with formal methods

- Extending Twitlang(er) to capture other specific features of Twitter e.g., direct messages, blocking of an account

- Investigating the synthesis/mining of Twitlang models from real Twitter accounts data
Thank you!

For further details about Twitlang and Twitlanger, visit

http://sysma.imtlucca.it/tools/twitlanger/