Balancing smartness and privacy

for the Ambient Intelligence

Harold van Heerde

Electrical Engineering, Mathematics and Computer Science
University of Twente

Ambient Intelligence....

... is everywhere and has influence on many aspects of daily life

... is invisible

... renders fast technical developments in new, powerful, sensing capabilities

... has a memory

Why does the AmI have this 'memory'?

Storing of context data to facilitate 'smartness'

- Context histories make it possible to:
  - Infer, predict and learn
    - pattern recognition
    - inferring next location
    - predicting next events
    - ...*et cetera*
  - Share knowledge
    - profile matching
    - finding experts
    - interests sharing
    - ...*et cetera*

Smartness...

...depends on quality and quantity of context data

Why does the AmI have this 'memory'?

Storing of context data to facilitate 'smartness'

- Context histories make it possible to:
  - Infer, predict and learn
    - pattern recognition
    - inferring next location
    - predicting next events
    - ...*et cetera*
  - Share knowledge
    - profile matching
    - finding experts
    - interests sharing
    - ...*et cetera*

Smartness...

...depends on quality and quantity of context data

This introduces a privacy problem!

- People will not always be aware of being monitored
- People do not know what happens or will happen with their data
- It will sometimes be hard to detect privacy violations
- Privacy sensitive ‘facts’ from the past will be kept forever in the system
  - Facts from the past can be used against you

Problem

We want to find a balance between smartness and privacy

This introduces a privacy problem!

- People will not always be aware of being monitored
- People do not know what happens or will happen with their data
- It will sometimes be hard to detect privacy violations
- Privacy sensitive ‘facts’ from the past will be kept forever in the system
  - Facts from the past can be used against you

Problem

We want to find a balance between smartness and privacy

Why privacy protection?

- Data should not be retained excessively
  - Imposed by the law (U.N. regulations for example)
    - Reduces impact of hacking
    - Avoid tracing
  - Trusted organizations are pushed to respect the law

Related work

- *k*-Anonymization, l-diversity, et cetera
- Hippocratic databases, p3p policies
- Access control (encryption, micro-views, et cetera)
- ...

Why privacy protection?

- Data should not be retained excessively
  - Imposed by the law (U.N. regulations for example)
    - Reduces impact of hacking
    - Avoid tracing
  - Trusted organizations are pushed to respect the law

Related work

- *k*-Anonymization, l-diversity, et cetera
- Hippocratic databases, p3p policies
- Access control (encryption, micro-views, et cetera)
- ...

Table of contents

Architecture

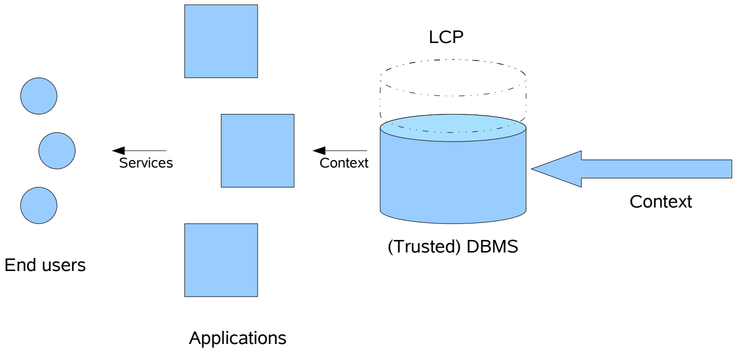


Figure: *Architecture*

Assumptions

- DBMS - Honest
  - DB and DB admin can be trusted *now* and in the near future, but might become untrusted in the future.
- Applications - Trusted
  - Applications have an interest to respect privacy (in order to keep their market segment).
  - Application code and data exchange communication cannot be attacked
- End users - Malicious
  - Only results or services given by applications are visible for end users.
  - Attacks because of (physical) spying between users are not addressed.

Assumptions

- DBMS - Honest
  - DB and DB admin can be trusted *now* and in the near future, but might become untrusted in the future.
- Applications - Trusted
  - Applications have an interest to respect privacy (in order to keep their market segment).
  - Application code and data exchange communication cannot be attacked
- End users - Malicious
  - Only results or services given by applications are visible for end users.
  - Attacks because of (physical) spying between users are not addressed.

Assumptions

- DBMS - Honest
  - DB and DB admin can be trusted *now* and in the near future, but might become untrusted in the future.
- Applications - Trusted
  - Applications have an interest to respect privacy (in order to keep their market segment).
  - Application code and data exchange communication cannot be attacked
- End users - Malicious
  - Only results or services given by applications are visible for end users.
  - Attacks because of (physical) spying between users are not addressed.

Access control or limited retention?

Access control

- Policies define until when and to which data applications have access.
- Data is kept and protected within the system
- The DB (and administrator) should be trusted now and in the future

Limited retention

- Use policies to define until when and which data is kept in the system
- Physical removal of data
- DB can be trusted *now*, but might be not in the future

Access control or limited retention?

Access control

- Policies define until when and to which data applications have access.
- Data is kept and protected within the system
- The DB (and administrator) should be trusted now and in the future

Limited retention

- Use policies to define until when and which data is kept in the system
- Physical removal of data
- DB can be trusted *now*, but might be not in the future

Progressive degradation (as a retention model)

- Fill the gap between ‘all or nothing’
  - Destroy all data means no smartness for applications
  - Keeping all data means possible privacy violation for users

Example

- A supermarket uses accurate data to predict if new cash desks should be opened
- It uses per customer buying information to make personalized advertisements
- And uses general statistics to optimize its selection of goods.

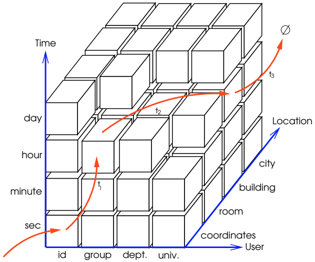
The Life-Cycle Policy model

Progressive value degradation of context histories

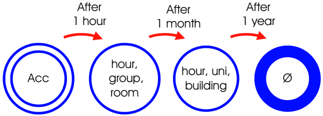
- Users specify when and how data should be degraded
- Goal is to protect context histories:
  - Degraded data is less privacy sensitive
- Current and recent accurate data is still available
  - Degraded data is still useful in terms of smartness
- There is room for negotiation about policies between application and users

The LCP model

- Data modeled as context triplets
- Triplets are elements in different states of accuracy
- Life-Cycle policies are transitions between states
- Users can specify their own LCP



(a) The Cube



(b) Example LCP

Functional degradation

- Data processing can be translated to SQL statements
- ‘Disable’ SQL operators by transforming the data
- Progressive degradation of the accuracy of those *abilities*

Example

- Keep the join ability without keeping real time values
- Degrade the ability from ‘join on minutes’ to ‘join on hour’

Functional degradation

- Data processing can be translated to SQL statements
- ‘Disable’ SQL operators by transforming the data
- Progressive degradation of the accuracy of those *abilities*

Example

- Keep the join ability without keeping real time values
- Degrade the ability from ‘join on minutes’ to ‘join on hour’

Value degradation and functional degradation

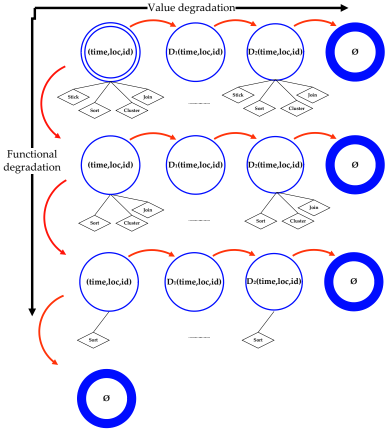


Figure: *(Natural) value degradation, and functional degradation of abilities*

Degradation in isolation for a known query set

Application oriented approach

Adequacy

Given a query Q on a dataset D, there is a degradation function V such that an alternative query Q' on $V(D)$ can be found which produces the same result as Q.

- Provide an adequate degradation function for a known set of queries
- Goal is to keep the least amount of information necessary to provide adequacy

Example: join on time

Query: $\pi_{door.event,window.event}\left(Door \bowtie_{hour(time)} Window\right)$

event	time	loc	id
e1	15.10	3090	1
e2	15.10	3090	2
e3	15.15	2045	2
e4	16.30	4180	1
e5	17.00	3090	3
e6	17.15	5360	3

(a) Door

event	time	loc	id
w1	15.10	3090	2
w2	16.05	1027	1
w3	17.02	2045	3
w4	18.05	2045	3
w5	18.15	2045	3
w6	18.16	2045	1

(b) Window

Result: $\{(e_1, w_1), (e_2, w_1), (e_3, w_1), (e_4, w_2), (e_5, w_3), (e_6, w_3)\}$

Example: join on time (cont'd)

Query: $\pi_{V(door).event', V(window).event'}(V(Door) \bowtie_{time'} V(Window))$

event'	time'	loc'	id'
e1	a	3090	1
e2	a	3090	2
e3	a	2045	2
e4	b	4180	1
e5	c	3090	3
e6	c	5360	3

(c) V(Door)

event'	time'	loc'	id'
w1	a	3090	2
w2	b	1027	1
w3	c	2045	3
w4	d	2045	3
w5	d	2045	3
w6	d	2045	1

(d) V(Window)

$18 \mapsto d$

Result: $\{(e_1, w_1), (e_2, w_1), (e_3, w_1), (e_4, w_2), (e_5, w_3), (e_6, w_3)\}$

Example: join on location

Query: $\pi_{door.event,window.event}\ (Door \bowtie_{loc} Window)$

event	time	loc	id
e1	15.10	3090	1
e2	15.10	3090	2
e3	15.15	2045	2
e4	16.30	4180	1
e5	17.00	3090	3
e6	17.15	5360	3

(e) Door

event	time	loc	id
w1	15.10	3090	2
w2	16.05	1027	1
w3	17.02	2045	3
w4	18.05	2045	3
w5	18.15	2045	3
w6	18.16	2045	1

(f) Window

Result: $\{(e_1, w_1), (e_2, w_1), (e_3, w_3), (e_3, w_4), (e_3, w_5), (e_3, w_6)\}$

Example: join on location (cont'd)

Query: $\pi_{V(door).event', V(window).event'}(V(Door) \bowtie_{loc'} V(Window))$

event	time	loc	id
e1	15.10	a	1
e2	15.10	a	2
e3	15.15	b	2
e4	16.30	d	1
e5	17.00	a	3
e6	17.15	e	3

(g) Door

event	time	loc	id
w1	15.10	a	2
w2	16.05	c	1
w3	17.02	b	3
w4	18.05	b	3
w5	18.15	b	3
w6	18.16	b	1

(h) Window

$3090 \mapsto a, 2045 \mapsto b, 1024 \mapsto c, 4180 \mapsto d, 5360 \mapsto e$

Result: $\{(e_1, w_1), (e_2, w_1), (e_3, w_3), (e_3, w_4), (e_3, w_5), (e_3, w_6)\}$

Problems

- Additional information is needed to keep the degraded data *adequate*
  - Possibly use secure hardware, access control, distributed keys, *et cetera*
  - At least the data itself can be stored in a non-secure database

Possible approach: one-way hash function

- But, domain is finit and sometimes small: easy to brake irreversability
- Use an additional key, and keep the key secret
  - Still requires access control, and disclosure of the key means disclosure of all data

Problems

- Additional information is needed to keep the degraded data *adequate*
  - Possibly use secure hardware, access control, distributed keys, *et cetera*
  - At least the data itself can be stored in a non-secure database

Possible approach: one-way hash function

- But, domain is finit and sometimes small: easy to brake irreversability
- Use an additional key, and keep the key secret
  - Still requires access control, and disclosure of the key means disclosure of all data

Conclusion

Ultimate goal is to balance privacy and smartness:

- Giving control to users limits asymmetric information
- Physical removal of data prevents unauthorized data disclosure
- Progressive degradation balances users wishes and application requirements

Questions?