UNIVERSITY OF Southampton

School of Electronics and Computer Science

Tracing *where* and *who* provenance in linked data - a calculus -

http://id.ecs.soton.ac.uk/person/9724

with M. Dezani-Ciancaglini and R. Horne

UNIVERSITY OF
Southampton
School of Electronics
and Computer Science

Tracing *where* and *who* provenance in linked data - a calculus -

http://id.ecs.soton.ac.uk/person/9724

Vladimiro Sassone

with M. Dezani-Ciancaglini and R. Horne

A six-slides talk featuring:

UNIVERSITY OF Southampton
Electronics and Computer Science

- an operational model of linked data processes that produce, consume, publish and query linked data;
- a model of *where* and *who* (and perhaps *why*) provenance for linked data;
- a *logic* for provenance patterns to specify *queries* over linked data
- a typing system to enforce *access control* policies based on provenance patterns and local policies.

Syntactic models

UNIVERSITY OF Southampton
Electronics and Computer Science

Or: why do I think we should be using more of them

Syntactic models

UNIVERSITY OF Southampton
Electronics and Computer Science

Or: why do I think we should be using more of them

Short answer: cause they are very good

Syntactic models

UNIVERSITY OF Southampton
Electronics and Computer Science

Or: why do I think we should be using more of them

Short answer: cause they are very good
Longer answer: cause they are very, very good

Syntactic models

UNIVERSITY OF Southampton
Electronics and Computer Science

Or: why do I think we should be using more of them

Short answer: cause they are very good
Longer answer: cause they are very, very good

They are at the same time *syntax* and *semantics*: provide unambiguous, compositional semantics and abstract/logical/(co)inductive reasoning techniques;

yet, they are executable, behavioural, easy to handle, compact, essential, and come with algebraic and type theories which support simple syntactic proofs, possibly formalisable in automated theorem provers.

Provenance & Patterns

UNIVERSITY OF Southampton
Electronics and Computer Science

agents: $\alpha, \beta \ldots$; locations: $\ell, m, \ldots$; functions: $f, g, \ldots$

Provenance & Patterns

UNIVERSITY OF Southampton
Electronics and
Computer Science

agents: $\alpha, \beta \ldots;$ locations: $\ell, m, \ldots;$ functions: $f, g, \ldots$

patterns: *agents* $\sigma ::= \alpha \mid \_$ *locations* $\Lambda ::= \ell \mid a \mid \_$

function tags $\phi ::= f \mid \_$

Provenance & Patterns

UNIVERSITY OF Southampton
Electronics and Computer Science

agents: $\alpha, \beta \ldots$; locations: $\ell, m, \ldots$; functions: $f, g, \ldots$

patterns: *agents* $\sigma ::= \alpha \mid \_$ *locations* $\Lambda ::= \ell \mid a \mid \_$

function tags $\phi ::= f \mid \_$

provenances:

$p ::=$		ϵ	empty
	$\mid$	(α, ℓ)	who-where
	$\mid$	$f\#$	why
	$\mid$	$p \cdot p$	concatenation
	$\mid$	$p \vee p$	disjunction

Provenance & Patterns

UNIVERSITY OF Southampton
Electronics and Computer Science

agents: $\alpha, \beta \ldots;$ locations: $\ell, m, \ldots;$ functions: $f, g, \ldots$

patterns: *agents* $\sigma ::= \alpha \mid \_$ *locations* $\Lambda ::= \ell \mid a \mid \_$

function tags $\phi ::= f \mid \_$

provenances:

$p ::=$		ϵ	empty
	$\mid$	(α, ℓ)	who-where
	$\mid$	$f\#$	why
	$\mid$	$p \cdot p$	concatenation
	$\mid$	$p \vee p$	disjunction

$$(Goble, \ell_1) \cdot (Buneman, \ell_2) \cdot f\# \cdot ((Moreau, \ell_1) \vee (Groth, \ell_0))$$

Provenance & Patterns

UNIVERSITY OF Southampton
Electronics and Computer Science

agents: $\alpha, \beta \ldots$; locations: $\ell, m, \ldots$; functions: $f, g, \ldots$

patterns: *agents* $\sigma ::= \alpha \mid \_$ *locations* $\Lambda ::= \ell \mid a \mid \_$

function tags $\phi ::= f \mid \_$

provenances:

$p ::=$		ϵ	empty
	$\mid$	(α, ℓ)	who-where
	$\mid$	$f\#$	why
	$\mid$	$p \cdot p$	concatenation
	$\mid$	$p \vee p$	disjunction

patterns:

$\pi ::=$		ϵ	empty
	$\mid$	(σ, Λ)	who-where
	$\mid$	$\phi\#$	why
	$\mid$	$\pi \cdot \pi$	concatenation
	$\mid$	$\pi \vee \pi$	disjunction
	$\mid$	${}^*\pi$	Kleene star
	$\mid$	$\top$	top

$$(Goble, \ell_1) \cdot (Buneman, \ell_2) \cdot f\# \cdot ((Moreau, \ell_1) \vee (Groth, \ell_0))$$

Provenance & Patterns

UNIVERSITY OF Southampton
Electronics and
Computer Science

agents: $\alpha, \beta \ldots$; locations: $\ell, m, \ldots$; functions: $f, g, \ldots$

patterns: *agents* $\sigma ::= \alpha \mid \_$ *locations* $\Lambda ::= \ell \mid a \mid \_$

function tags $\phi ::= f \mid \_$

patterns:

empty

-where

why

enation

inction

ne star

$\mid \quad \top$ top

$\mid \quad p \vee p$ disjunction

- $*\,[(\alpha, \_) \vee (\beta, \_)]$ only α and β ever wrote it.
- $\top\,(\_, \ell)$ the data originated in ℓ, with $\top = *(\_)$
- $\top\,(\beta, \_)\,(\alpha, \ell)\,\top$ data in ℓ by α in was reposted by β.
- $\neg\,(\top\,(\alpha, \_)\,\top)$ α never wrote the data anywhere.

$$(Goble, \ell_1) \cdot (Buneman, \ell_2) \cdot f\# \cdot ((Moreau, \ell_1) \vee (Groth, \ell_0))$$

Data, Expressions & Queries

UNIVERSITY OF Southampton
Electronics and Computer Science

stored data:

D ::=	$(\ell\ \ell\ \ell)^p$	tracked triple
|	$D \parallel D$	composition
|	0	empty data

expressions:

e ::=	x	variable
|	D	data
|	$f(e)$	application
|	$e \parallel e$	composition

Data, Expressions & Queries

UNIVERSITY OF Southampton
Electronics and Computer Science

stored data:

D	$::=$	$(\ell\,\ell\,\ell)^p$	tracked triple
	$\mid$	$D \parallel D$	composition
	$\mid$	0	empty data

expressions:

e	$::=$	x	variable
	$\mid$	D	data
	$\mid$	$f(e)$	application
	$\mid$	$e \parallel e$	composition

queries:

Q	$::=$	$(\lambda\,\lambda\,\lambda)^\pi$	triple pattern
	$\mid$	$Q \oplus Q$	choose
	$\mid$	$Q \otimes Q$	tensor
	$\mid$	$\exists a.Q$	exists
	$\mid$	$*Q$	iteration

Data, Expressions & Queries

UNIVERSITY OF Southampton
Electronics and Computer Science

stored data:

D	$::=$	$(\ell\,\ell\,\ell)^p$	tracked triple
	$\mid$	$D \parallel D$	composition
	$\mid$	0	empty data

expressions:

e	$::=$	x	variable
	$\mid$	D	data
	$\mid$	$f(e)$	application
	$\mid$	$e \parallel e$	composition

$$\frac{p \Vdash \pi}{C^p \models C^\pi} \qquad \frac{D \models Q\{^\ell/_a\}}{D \models \exists a.Q}$$

$$\frac{D \models Q_0}{D \models Q_0 \oplus Q_1} \qquad \frac{D_0 \models Q_0 \quad D_1 \models Q_1}{D_0 \parallel D_1 \models Q_0 \otimes Q_1}$$

queries:

Q	$::=$	$(\lambda\,\lambda\,\lambda)^\pi$	triple pattern
	$\mid$	$Q \oplus Q$	choose
	$\mid$	$Q \otimes Q$	tensor
	$\mid$	$\exists a.Q$	exists
	$\mid$	$*Q$	iteration

Data, Expressions & Queries

UNIVERSITY OF Southampton
Electronics and Computer Science

stored data:

$$D ::= (\ell\,\ell\,\ell)^p \quad \text{tracked triple}$$
$$\mid\ D \parallel D \quad \text{composition}$$
$$\mid\ 0 \quad \text{empty data}$$

expressions:

$$e ::= x \quad \text{variable}$$
$$\mid\ D \quad \text{data}$$
$$\mid\ f(e) \quad \text{application}$$
$$\mid\ e \parallel e \quad \text{composition}$$

$$\frac{p \Vdash \pi}{C^p \models C^\pi} \qquad \frac{D \models Q\{^\ell/_a\}}{D \models \exists a.Q}$$

$$\frac{D \models Q_0}{D \models Q_0 \oplus Q_1} \qquad \frac{D_0 \models Q_0 \quad D_1 \models Q_1}{D_0 \parallel D_1 \models Q_0 \otimes Q_1}$$

queries:

$$Q ::= (\lambda\,\lambda\,\lambda)^\pi \quad \text{triple pattern}$$
$$\mid\ Q \oplus Q \quad \text{choose}$$
$$\mid\ Q \otimes Q \quad \text{tensor}$$
$$\mid\ \exists a.Q \quad \text{exists}$$
$$\mid\ {*Q} \quad \text{iteration}$$

$$\exists a.\exists b.(a\,p\,b)^{(\_,\ell)\cdot\top}$$

$$\exists a.\exists b.(a\,p\,b)^{(\alpha,m)\cdot(\_,\ell)\cdot\top}$$

Processes & Behaviours

UNIVERSITY OF Southampton
Electronics and Computer Science

processes:

P ::=	0		termination
$\mid$	$\texttt{get}(Q, x).P$		consume
$\mid$	$\texttt{del}(Q, x).P$		delete
$\mid$	$\texttt{ins}(\lambda, e).P$		publish
$\mid$	$P + P$		choose
$\mid$	X		process variable
$\mid$	$\texttt{rec}X.P$		recursion
$\mid$	$\exists a : \texttt{Loc}(\mathcal{R}, \mathcal{D}, \mathcal{I}).P$		select location

Processes & Behaviours

processes:

P ::=	0		termination
|	$\texttt{get}(Q, x).P$		consume
|	$\texttt{del}(Q, x).P$		delete
|	$\texttt{ins}(\lambda, e).P$		publish
|	$P + P$		choose
|	X		process variable
|	$\texttt{rec}X.P$		recursion
|	$\exists a : \texttt{Loc}(\mathcal{R}, \mathcal{D}, \mathcal{I}).P$		select location

ignore this for the moment

Processes & Behaviours

processes:

P	::=	0	termination
	$\mid$	$\mathtt{get}(Q, x).P$	consume
	$\mid$	$\mathtt{del}(Q, x).P$	delete
	$\mid$	$\mathtt{ins}(\lambda, e).P$	publish
	$\mid$	$P + P$	choose
	$\mid$	X	process variable
	$\mid$	$\mathtt{rec}X.P$	recursion
	$\mid$	$\exists a : \mathtt{Loc}(\mathcal{R}, \mathcal{D}, \mathcal{I}).P$	select location

ignore this for the moment

systems:

S	::=	$\mathbf{0}$	termination
	$\mid$	D	stored data
	$\mid$	$\alpha[P]$	tagged process
	$\mid$	$S \parallel S$	interleave

Processes & Behaviours

processes:

$P ::=$		0	termination
	$\mid$	$\mathtt{get}(Q, x).P$	consume
	$\mid$	$\mathtt{del}(Q, x).P$	delete
	$\mid$	$\mathtt{ins}(\lambda, e).P$	publish
	$\mid$	$P + P$	choose
	$\mid$	X	process variable
	$\mid$	$\mathtt{rec}X.P$	recursion
	$\mid$	$\exists a : \mathtt{Loc}(\mathcal{R}, \mathcal{D}, \mathcal{I}).P$	select location

$$\frac{e \Downarrow C_1^{p_1} \parallel \cdots \parallel C_m^{p_m}}{\alpha[\mathtt{ins}(\ell, e).P] \longrightarrow \alpha[P] \parallel C_1^{(\alpha,\ell)\cdot p_1} \parallel \cdots \parallel C_m^{(\alpha,\ell)\cdot p_m}}$$

ignore this for the moment

systems:

$S ::=$		$\mathbf{0}$	termination
	$\mid$	D	stored data
	$\mid$	$\alpha[P]$	tagged process
	$\mid$	$S \parallel S$	interleave

Processes & Behaviours

UNIVERSITY OF Southampton
Electronics and Computer Science

processes:

P	$::=$	0	termination
	$\mid$	$\mathtt{get}(Q, x).P$	consume
	$\mid$	$\mathtt{del}(Q, x).P$	delete
	$\mid$	$\mathtt{ins}(\lambda, e).P$	publish
	$\mid$	$P + P$	choose
	$\mid$	X	process variable
	$\mid$	$\mathtt{rec}X.P$	recursion
	$\mid$	$\exists a : \mathtt{Loc}(\mathcal{R}, \mathcal{D}, \mathcal{I}).P$	select location

$$\frac{e \Downarrow C_1^{p_1} \parallel \cdots \parallel C_m^{p_m}}{\alpha[\mathtt{ins}(\ell, e).P] \longrightarrow \alpha[P] \parallel C_1^{(\alpha,\ell)\cdot p_1} \parallel \cdots \parallel C_m^{(\alpha,\ell)\cdot p_m}}$$

systems:

S	$::=$	$\mathbf{0}$	termination
	$\mid$	D	stored data
	$\mid$	$\alpha[P]$	tagged process
	$\mid$	$S \parallel S$	interleave

ignore this for the moment

$$\alpha[\mathtt{get}(\exists a.\exists b.(a\,p\,b)^{(\_,\ell)\cdot\top}, x).$$
$$\mathtt{del}(\exists a.\exists b.(a\,p\,b)^{(\alpha,m)\cdot(\_,\ell)\cdot\top}, y).$$
$$\mathtt{ins}(m, x).0]$$

Types & Access control policies

UNIVERSITY OF Southampton
Electronics and Computer Science

Site policy for Retrieval, Deletion and Insertion:

$$\ell : \texttt{Loc}(\mathcal{R}, \mathcal{D}, \mathcal{I})$$

provenance patterns describing allowed actions

Eg, insertion policy for Southampton might be:

$$\{\langle Groth, \top\rangle, \langle Buneman, (Moreau, Soton) \cdot \top\rangle\}$$

Then, for all D and C^p

$$\vdash Groth[\texttt{ins}(Soton, D).0]$$

$$Groth[\texttt{ins}(Soton, C^p).0] \to Groth[0] \parallel C^{(Groth,Soton)\cdot p}$$

Yet $\vdash Buneman[\texttt{ins}(Soton, D).0]$ only if the provenances of all of D are of the kind $(Moreau, Soton) \ldots$

Fancy more like this?

UNIVERSITY OF Southampton
Electronics and
Computer Science

If in your wisdom you want to see more of my work on linked data, please go to:

http://eprints.soton.ac.uk

and search "*sassone linked data*"

http://id.ecs.soton.ac.uk/person/9724