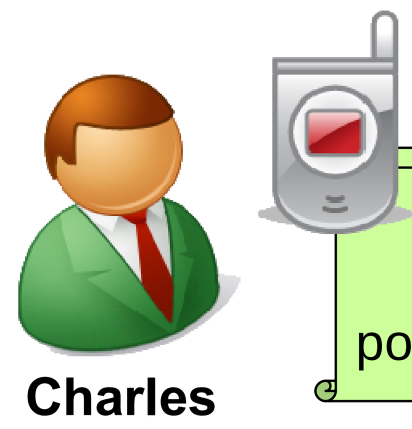
Towards Learning Privacy Policies

Arosha K Bandara
Department of Computing, The Open University
Walton Hall Campus, Milton Keynes MK7 6AA, UK
E-mail: a.k.bandara@open.ac.uk

Alessandra Russo and Emil C Lupu
Department of Computing, Imperial College London
South Kensington Campus, London SW7 2AZ, UK
E-mail: {a.russo, e.c.lupu}@imperial.ac.uk

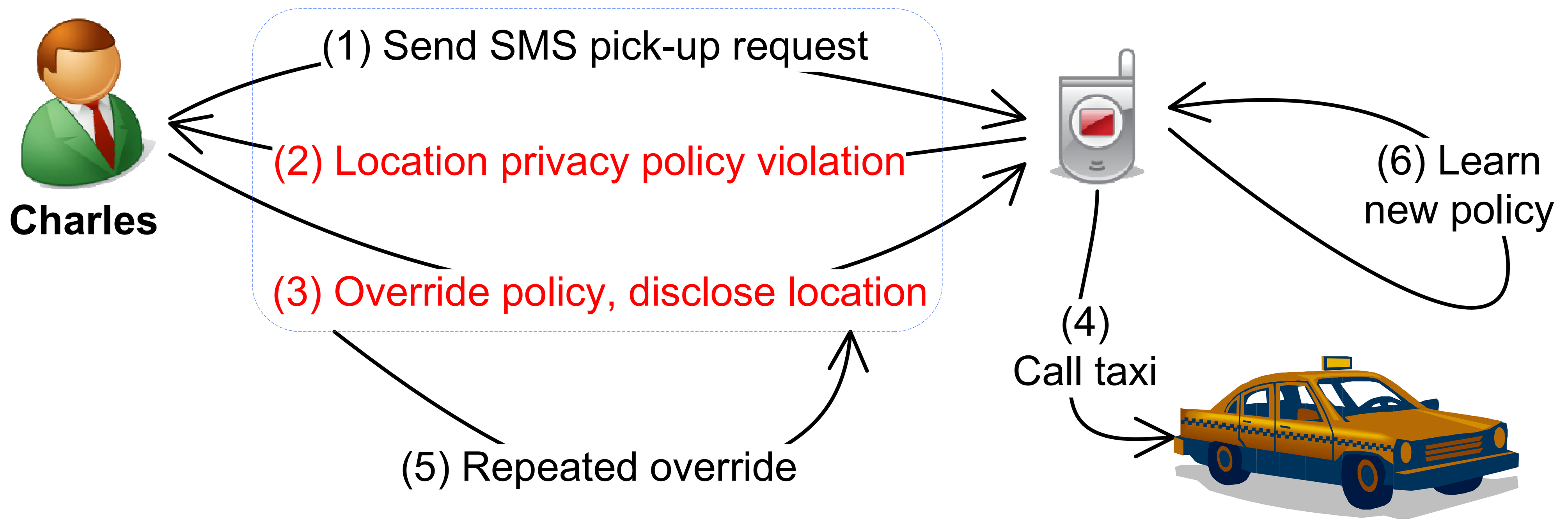
SCENARIO

Alice and Bob's son Charles is involved in many after-school activities. Concerned for his safety whilst travelling to and from these activities, Charles' parents buy him a new mobile phone that has a GPS tracking feature together with a Privacy Manager (PM) tool. To prevent Charles from unintentionally disclosing is location to others, Bob configures the PM with a policy that states that only Alice and Bob can read Charles' location information.



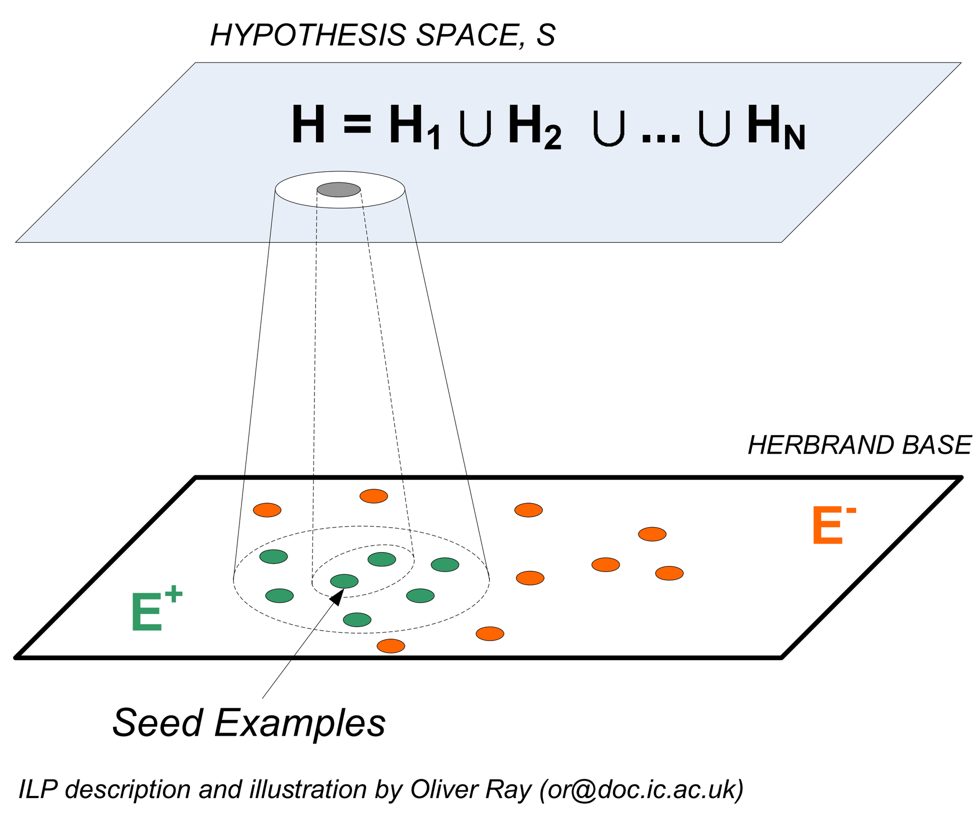
EXAMPLE PRIVACY POLICY

parent_of(charles, alice), parent_of(charles, bob),
∀X: parent_of(charles, X) →
policy(allow, X, read, charles, location, 'have peace of mind')



One day Charles needs a lift home and uses a taxi firm, `zCar', that allows customers to send SMS requests containing their location (1). However, when Charles tries to send a pick-up request, his PM informs him that this would violate his location privacy policy (2). Charles chooses to override his policy (3) and soon a taxi arrives to take him home (4). The next time Charles needs a lift, he uses another firm offering the same service,`qCab', and is again forced to override his policy (5). Over time, Charles' PM learns this behaviour and suggests a new policy that will disclose his location to taxi firms whenever he requests a pick-up (6).

INDUCTIVE LOGIC PROGRAMMING



Inductive Logic Programming (ILP) performs the following computational task:

Given:
B Background theory: *Set of Horn clauses*
E^+ Positive examples: *Set of ground atoms*
E^- Negative examples: *Set of ground atoms*
S Hypothesis space: *Set of Horn clauses*

Find:
H Hypotheses, $H \subseteq S$: *Set of Horn clauses*

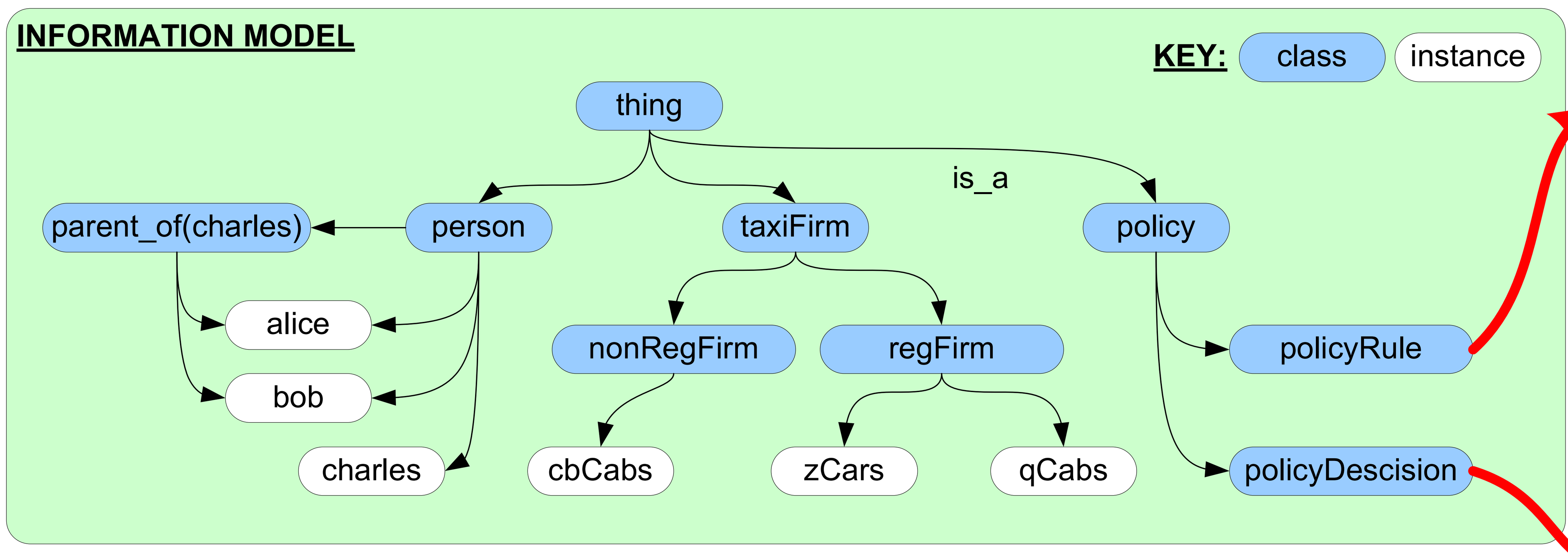
Such that:
$H \cup B \models e^+$ for all $e^+ \in E^+$
$H \cup B \not\models e^-$ for all $e^- \in E^-$

Our approach to learning privacy policies advocates using Inductive Logic Programming (ILP) over statistical techniques because ILP produces rules (privacy policies) that are comprehensible to the end user and at the same time amenable to automated analysis.

In our formalisation, we map the privacy policy information to the types of information required by the ILP procedure in the following way:

Background Theory, B	:	The initial set of privacy policies. The information model for subjects, targets and actions.
Positive Examples, E+	:	The policy decisions made by the system or the user.
Negative Examples, E-	:	Invalid policy decisions, e.g. allow and deny disclosure of location.
Hypothesis Space, S	:	The schema for privacy policy rules

LEARNING PRIVACY POLICIES



The information model defines the relationships between the classes of object and object instances. This forms the background theory for the ILP procedure. For example, in the above diagram 'alice' is an instance of the 'person' class and 'regFirm' is a subclass of 'taxiFirm'.

These relationships are formally encoded using the predicate *is_a(X, Y)* – denoting that X is a subclass/instance of Y. Therefore, the given examples would be encoded as: *is_a(alice, person)* and *is_a(regFirm, taxiFirm).*

The hypothesis space for the ILP procedure is defined by rules with *policy(…)* predicates in the head and *is_a(…)* predicates in the body.

parent_of(charles, alice), parent_of(charles, bob),
∀X: parent_of(charles, X) →
policy(allow, X, read, charles, location, 'have peace of mind')

Policy rules are formally encoded using the *policy(…)* predicate. The above example encodes the policy that if the Subject *X* is a parent of Charles, then X is allowed to *read* Charles' *location* data for the purpose *'have peace of mind'*

policy(allow, zCars, read, charles, location, 'pickup')
policy(allow, qCabs, read, charles, location, 'pickup')

Policy decisions are also encoded using the *policy(…)* predicate. The above example encodes the decisions given in the above scenario where Charles overrides his policy to allow *zCars* and *qCabs* to read his *location* data for the purpose *'pickup'*. These decisions are the positive examples, E^+, given to the ILP procedure

is_a(X, regFirm) → policy(allow, X, read, charles, location, 'pickup')

For the example scenario, the ILP procedure uses the information model and example policy decisions to learn the above rule that states that any subject X which is a *regFirm* should be allowed to read Charles' *location* data for the purpose *'pickup'.*

References:
[1] P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter. *EPAL 1.1*. URL: http://tinyurl.com/35xpon,(1.10. 2003).
[2] L. Cranor. *Web Privacy with P3P*. O'Reilly, USA, 2002.
[3] S. Muggleton and C. Bryant. Theory completion using inverse entailment. In *10th Int. Conf. on Inductive Logic Programming*, pages 130–146, 2000.
[4] O. Ray, K. Broda and A. Russo. A Hybrid Abductive Inductive Proof Procedure. *Logic Journal of the IGPL* 12(5):371-397, 2004.