A Normative Approach to Exploring Multi-Agency Privacy and Transparency

Julian Padget[1]*, Ken Satoh[2], and Fuyuki Ishikawa[3]

[1] University of Bath, Dept. of Computer Science, UK
j.a.padget@bath.ac.uk
[2] National Institute of Informatics/Sokendai, Japan
ksatoh@nii.ac.jp
[3] National Institute of Informatics, Japan
f-ishikawa@nii.ac.jp

Abstract. Privacy and transparency issues straddle a grey area that encompasses data protection legislation, legally binding confidentiality agreements, service contracts and social conventions. This very variety can easily result in the issues being viewed as "someone else's problem", because the domain obscures the commonalities between the events and states that characterize the risks to privacy and the complementary requirements for transparency, regardless of context.
Against this backdrop, we investigate how a data subject (organization or individual) may become aware of a vulnerability to their data, whether it is directly under their control or, more likely only indirectly through the force of obligations on third parties, subject to law, contract or convention. Ideally, we seek to recognize the eventuality of a vulnerability before the damage is done, rather than thanks to a notification as a consequence of the transparency policy.
Our approach uses a general-purpose event-based modelling framework in which relevant elements of regulations and conventions can be encoded so that we may examine the interplay between them as subjects go about their business. As a result, it becomes possible to experiment with and revise the rules that govern behaviour in order to evaluate alternative approaches to the balance between legal and social action and the management of privacy and transparency.

1 Introduction

The use of mobile phones, or more specifically the many apps they now support, allows many organizations to collect data about individual activities. In isolation, such data may not be especially useful, but in combination, it is perceived as a significant potential threat to indivdual privacy. The notion of transparency in general, as a means to expose the actions and performance of organizations [13, 22] and in particular, as the basis for constructing a personalized log [15] of data subject transactions, is seen as a potential source of protection for the individual.

To be more specific, software platform mediated interactions are increasing rapidly – primarily in the form of social networking, but increasingly also (government) service provision and on-line purchasing – leading to the capture, unintended or otherwise, of

* Partially supported by the visiting faculty programme at NII, July 2013

huge amounts of personal data and the effective entanglement of human and software actions, creating so-called socio-technical systems in which both humans and software are pro-active. Such systems often operate at a global scale, meaning that principle the interactions and the data are governed by numerous national laws regarding data protection, privacy and transparency (amongst others), sometimes even different aspects of the same interaction may technically occur in different jurisdictions. This diversity of software and regulation (national and organizational) makes both the development and the proper validation of software for deployment context-dependent compliance extremely problematic. We believe that the solution to this problem lies in making software systems that are situationally aware and can adapt to meet the (non-functional) requirements of their deployment through being able to process and reason about matters such as the applicable legislation and organizational policies, through continuous validation.

Thus, the subject of this paper is the modelling of the issues that arise from the shared contexts created by the interaction of humans, organizations and multiple software platforms. We illustrate our approach by considering the matter of detecting certain situations within software systems – for example: business processes, social networking platforms – and how idealizations of principles of policy, user contract or law might affect, both statically and dynamically, the compliance of such software systems with, in this particular case, requirements for security, privacy and transparency. The classification of these features as "non-functional" has in the past been a way of moving them out of the implementation scope of the software system and putting responsibility for verifying compliance on to some form of (often manual) audit process that may be applied to the code as part of the acceptance phase or in the form of checks of system logs at regular or irregular intervals in the live phase.

It is becoming apparent however, that the system frontier can be extended to encompass non-functional requirements in certain forms and there is increasing recognition of the attraction of the representation of "requirements at run time", which although first expressed nearly two decades ago [12], it seems that it is only relatively recently that a research agenda has started emerging in the software engineering community [28], driven in part by the demands of adaptive systems [3] and the possibilities offered by techniques such as argumentation [30].

In parallel, the concept of norm was being refined and formalized through research in logic [1], legal reasoning, social sciences [23] and artificial intelligence [19], leading to computational models of normative systems that capture notions of required behaviour. Such models could be analysed in their own right for intended or unintended outcomes, utilized in agent-based simulation to guide agent behaviour, provide more socially-plausible non-player character behaviour in computer (serious) games and eventually to inform the actions of intelligent agents in socio-cognitive systems.

The purpose of these very selective histories is to support the observation that some degree of convergence is appearing between the strands of research on software engineering and on normative modelling and reasoning, and hence to argue that the representation of policy and processes of reasoning about policy can become part of software systems and begin to address the significant concerns that surround how the agility and reach of global software platforms impacts an individual's privacy.

In the next section we survey a range of related work covering (i) access control and how it is beginning to address the problem of data that is beyond direct control (ii) the implications of transparency research (outside the computer science domain) for software systems, and (iii) mechanisms for long-term association of data and policy. In section 3, we examine the interplay of privacy, security and transparency to establish some context for the illustrative scenario that follows. Hence, in section 4, after a brief introduction to the modelling framework, we outline a scenario involving mobile phones, personal data and social networking platforms and show how the model can expose situations of interest to designers and policy makers, as well as allowing experimentation with (compliance) mechanisms to help in meeting implementation, policy and possibly even legislative goals. We conclude with some discussion (section 5) and consideration of the (many) remaining challenges.

2 Related Work

Unfortunately, one characteristic of privacy issues is that breaches are by their very nature only recognized after the fact, while risks to privacy have a tendancy to be viewed as vague, low probability events, until they happen. Consequently, this has lead to a concentration of effort on security, epitomised by the Role-Based Access Control (RBAC) [24] family of work that aims to prevent unauthorized access to data. The problem is that it is a pre-requisite of much interaction that data must be transmitted from one agency to another, which typically results in the data being outside the scope of the access control policy that originally secured it, which Park et al. [24] acknowledge: "UCON systems are likely to be implemented and managed under the control of one of the three subject sides: consumer, provider, or identifiee. This implies it's hard to guarantee availability of adequate control mechanisms implemented for the other two sides on the rights and usage of rights.". RBAC-oriented approaches have lately [6] begun to consider such scenarios, in the context of social networks, in which access may be controlled by the fusion of multiple authorization policies (pertaining to the data target and the accessor), such as the scenario of an individual being tagged in a photo uploaded by another individual. Despite the increasing richness however, the scenario does remain inside a single domain, as determined by the particular social networking platform.

Thus, both the problems of data beyond the control of the subject and of post-facto discovery of breaches remain. Risk-aware RBAC aims reduce the occurrence of the former by attention to the lattter in that it proposes *a priori* assignment of risk valuations to actions. The risk of granting a request is defined as the combination of the cost of misuse and the likelihood of misuse. The fundamental problem with the approach is expressed in the phrase "system administrators have "valued" the cost of ... misuse" (sic) [4]. Actual risk mitigation strategies are defined in terms of interval maps to system and user obligations [5], but while the first is directly enforceable, the second may only be indirectly, either through the threat of social sanctions (loss of trust, ostracism) or legal process subject to the availability of evidence. This last, picks up the thread from the introduction and begins to highlight the role that transparency has to play.

At first sight, transparency seems like monitoring, since from a functional point of view, the purpose is to deliver accurate and comprehensible (requirements from [13]) information about what is happening by means of which organizational processes to data pertaining to the subject (of the data). But as we discuss in the next section, it is more than that, since the transparency *policy* must define what (how and why) data is to be reported, how the policy satisfies the encompassing legislative framework, the sanctions applicable in the case of non- or mis-reporting and the interconnection with other regulatory mechanisms (inside and outside the organization, to independent regulatory authorities, ombudsmen, etc.).

However, the matter of data that has gone beyond the reach of the originating system, remains problematic. In the case of risk-aware RBAC above [5], the system obligations can lead to the transformation of data before access is permitted (e.g. on-the-fly anonimization), while user obligations take the form of actions that must be carried out (e.g. obtaining *post facto* authorization for a time critical operation) or result in restriction on behaviour in the future. Karjoth et-al [17] propose a cross-enterprise platform, focussing on the data itself, with which "sticky policies" are associated, so that actions in whichever organization is processing the data are mediated by a policy engine that interprets the data's associated policy. This appears to constitute a single point of failure, but could perhaps be supplied by a replicated service in practice.

3 Privacy, Security and Transparency

The definition of transparency seems to be open to debate, indeed it is more often described than defined. For the purposes of this paper, we start from the quite general position, advanced in "Full Disclosure" [13], which considers transparency in the broadest (non-informatics) societal setting, from which we put forward "the ability to obtain information about a process in an organization". Such a definition is not especially helpful precisely because of its lack of specificity. But if we sharpen the focus to the individual, which will be the subject of our later case study, then we can formulate the problem more concisely as "somewhere, some process is doing something with data that I have provided" and the application of transparency should permit an individual to establish:

1. Where this is happening
2. What is the intention
3. What occurred, and
4. What are the outcomes.

It is perhaps worth emphasizing, that it is not just what was done with the data, but also the *location* – since this affects the jurisdiction – the *motivation* – the intention may have been legal, but what happened was not – and the *consequences* – particularly in the case of some kind of failure at the previous stage, may have undesirable results – that all contribute to the delivery of transparency.

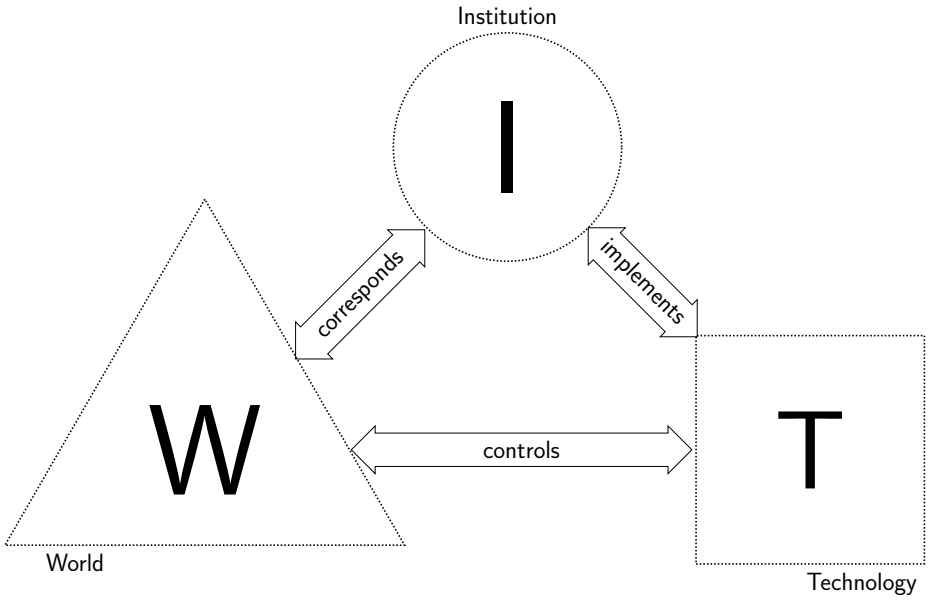

Fig. 1. The physical **W**orld, a governing **I**nstitution and mediating **T**echnology

3.1 The role of institutions

Our preliminary conceptualization of the problem and how to approach its solution appear in Figure 1[4]. We consider the real world (W) and the technology world (T) to be tied together by the actions of human and software agents[5]. However, in the background to this play are normative states and rules, some created by social conventions, emerging from the interaction of the agents [26, 27], some defined through social processes, such as organizations or legislative processes, gathered into an idealized (institutional) model. Such states, and the rules that are intended help maintain or avoid them, are known as institutions [21, 14] and, as intimated, may be either implicit in the mental states of the actors or explicit in the form of procedure manuals or laws, or as is the case in section 4, as a specification grounded in computational logic.

The drivers for the WIT model are the actions of the human and software participants. Unavoidably, this world is only partially observable (not that fully observable is desirable, but the consequence is that decisions must be made under uncertainty because of incomplete information) because not every human action is detectable, and even if it were, the interpretation is going to be questionable, while most but not all – simply because of implementation choices – software actions can be logged. This latter deluge of data creates its own problems simply through volume, which underlines the value of the idealization represented in the institution, such that only empowered relevant actions –

[4] An earlier version of this diagram appears in [20]

[5] We use the term agent, without predjudice, to refer to any reactive or pro-active software component.

that affect the institutional state – are taken into account. The WIT model offers three perspectives that are each connected with the other: (i) W↔I establishes the so-called "counts-as" relationship [29, 16] which establishes the correspondence between W actions and I actions and the (institutional) consequences entailed, which in due course carry over to W, (ii) I↔T establishes the functionality of T such that it delivers the actions identified in I, and finally (iii) W↔T enables the interactions in W, by providing the inputs and outputs for the human and software agents in W.

There are numerous ways in which the I component can be implemented (see [11] for an overview of organizational and institutional modelling). The approach we take in section 4 has been described in detail in several papers (see for example [10]) with a formal model as outlined in Figure 2, comprising a set of facts that describe the current state of the institution and a pair of relations, whose actions are triggered by events:

1. We distinguish between external events and institutional events: the former are not controlled by the institution and act as triggers for institutional action; the latter can have permission and power associated with them to reflect whether an action is allowed at some time and whether the action (by some actor) has any institutional effect[6].
2. The generation relation $\mathcal{G}$, whose domain is an event and the current state of the institutions – so that processing is conditional upon the current state – and range is the set of events defined by the institution; hence, depending on context, a real world action is recognized as an institutional action (counts-as). Initally, $\mathcal{G}$ is invoked with an external event: if that event is recognized by the institution, it may generate an institutional event, which by the repeated application of $\mathcal{G}$ may in turn generate another and another, all of which are dependent on the current institutional state.
3. The consequence relation $\mathcal{C}$, whose domain is as for $\mathcal{G}$ and range is the set of facts denoting the institutional state. $\mathcal{C}$ is invoked for each of the events generated so firing any associated rule for the addition and/or deletion of (so-called inertial) facts from the institutional state.
4. Additionally, there are non-inertial facts, that are true only while some condition over the institutional state is true. These permit the recognition of situations whose constituent facts may be initiated (or terminated) by events at different points in time.

As a result, the single external event and all the institutional events are treated as occurring simultaneously and having a simultaneous effect on the institutional state (meaning the order in which events are generated within a single update has no effect). The institutional facts comprise the union of facts about the $\mathcal{D}$omain being modelled, $\mathcal{P}$ermissions and po$\mathcal{W}$ers associated with actions and $\mathcal{O}$bligations arising from actions.

Although at first sight the approach has similarities to the Event Calculus and to Situation Calculus, two differences are worth highlighting: (i) the use of a two-level event structure, where brute events generate (possibly multiple) institutional events (leading

[6] For example: it has meaning if the chair of a meeting says that business is finished, but is meaningless if someone other than the chair says so.

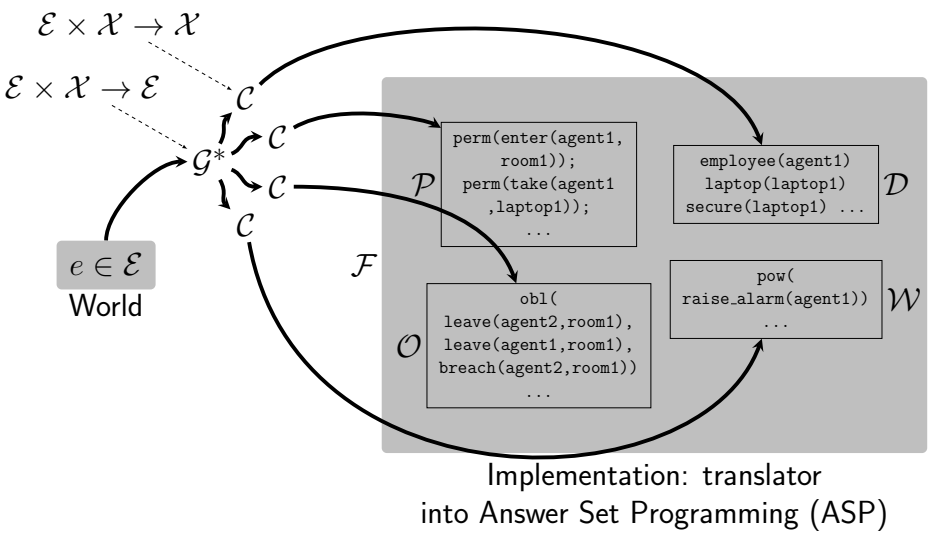

Fig. 2. Outline of formal model

to multi-institution models via inter-institutional events [9]) which initiate/terminate institutional facts, and (ii) the implementation in Answer Set Programming (rather than Prolog), which allows exploration of all possible traces over a finite number of steps [8], as a tool for design, or just single step evaluation, as an on-line institutional tracking tool [2, 18]. The justification and full development of the model appears in [7].

JP: Fig 2 restored from conference version, since we have more space

4 Phone data scenario

The scenario we have constructed to demonstrate the approach to policy exploration is based on the interaction of (smart)phones and social network platforms (SNPs). This offers a relatively accessible scenario, whose attractions and risks are widely perceived and which can be modelled at differing degrees of fidelity depending on objectives. Our purpose here is to present a scenario that is sufficiently complicated to demonstrate the interaction between phones, between phones and SNPs and between SNPs, where the actions of the transfer of data are governed by various kinds of policies, some of which whose violation may contravene privacy legislation.

The scenario description comes in two parts: a picture (see Figure 3) and a specification in the institutional modelling language InstAL [7]. The purpose of the latter is to provide a model that, through the application of an answer set solver, can be used to identify situations in which personal data is vulnerable *before* it is compromised, as well as situations in which it is compromised (in particular in the absence of consent, since this may constitute a breach of contract and consequently a potential legal violation). The model is used here as an off-line tool for the evaluation of policies, conditions and to identify situations of concern for designers, but it can in principle equally be used as

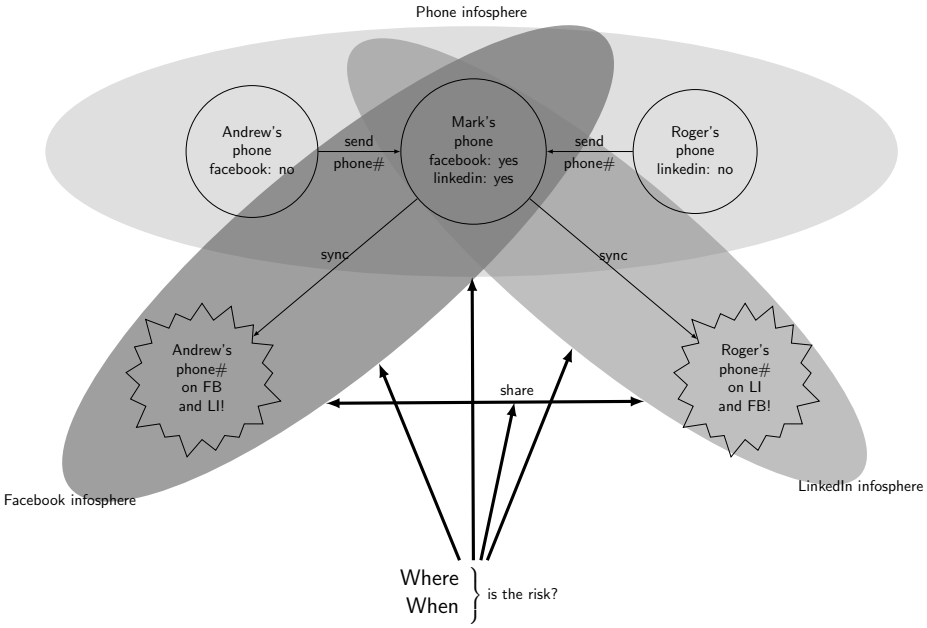

Fig. 3. The phone data scenario

an on-line tool to support intelligent agents [2] in simulation or in live applications. We illustrate the details of the scenario with some fragments of the InstAL specification.

The details are as follows:

1. Three communication spaces (labelled infospheres in Figure 3), for phones, for facebook and for linkedin, respectively[7].
2. Three phones, mediating the actions of three users (Andrew, Mark and Rodger) with one another and with SNPs, in some cases. The relationship between the phones is expressed using the `friend` relation:

```
1    friend(andrews_phone,marks_phone),
2    friend(marks_phone,andrews_phone),
3    friend(rodgers_phone,marks_phone),
```

3. Initially, each phone has its user's data stored on the phone. Additionally, Mark's phone has a facebook app and a linkedin app and (data) consent for both platforms, while Andrew's data (note: not phone) has consent only for linkedin and Rodger's has consent only for facebook:

```
1    consent(marks_data,facebook),
2    consent(marks_data,linkedin),
3    consent(andrews_data,linkedin),
4    consent(rodgers_data,facebook)
```

[7] Note: facebook and linkedin are just used as names; no particular criticism or properties should be ascribed to these choices.

4. The actions that occur are that Andrew and Rodger both communicate their data to Mark, he then synchronizes his data with facebook and linkedin and finally, the two SNPs share data with one another. Along the way, various situations concerning personal data are detected and flagged by facts in the model state, specifically:
   (a) A phone may only send data to a phone with which it has a friend relationship. While we cannot control physical world acts, in the sense that they are always empowered, we can express a policy over institutional acts to ensure adherence to the friend policy, thus:

```
1 exogenous event send(Resource,Resource);
2 inst      event iSend(Resource,Resource);
3 send(R1,R2) generates iSend(R1,R2);
4 iSend(R1,R2) initiates has(R2,R1);
5 perm(iSend(R1,R2)) when has(R3,R1), friend(R3,R2);
6 pow(iSend(R1,R2)) when has(R3,R1), friend(R3,R2);
```

   Lines 3–5 of which translate, respectively, as:

```
occurred(iSend(R1,R2),I) :- occurred(send(R1,R2),I),
   holdsat(pow(dataWatch,iSend(R1,R2)),I),
   resource(R1),
   resource(R2),
   instant(I).
```

```
initiated(has(R2,R1),I) :-
   occurred(iSend(R1,R2),I),
   holdsat(live(dataWatch),I),
   resource(R1),
   resource(R2),
   instant(I).
```

```
holdsat(perm(iSend(R1,R2)),I) :-
   holdsat(has(R3,R1),I),
   holdsat(friend(R3,R2),I),
   resource(R1),
   resource(R2),
   resource(R3),
   instant(I).
```

   which declares an external event and an institutional counterpart for sending, such that the external event generates the institutional event which initiates the fact `has(R2,R1)`, but subject to the policy expressed through the last two lines, in which the power and permission for `iSend`[8] are only true while the sender has the data in question and the `friend` relation is true.
   (b) The presence of a SNP app on a phone represents a risk to any data on the same device. We have chosen to model this issue with an opt-in sticky policy [17], so unless there is an explicit `consent` relation between some data and some platform, the data is `vulnerable`, as recognized by `situation1`:

```
vulnerable(R,P) when situation1(R,P);
situation1(R2,facebook) when
   typeOf(R1,phone), typeOf(R2,data),
   has(R1,R2), has(R1,facebook_app),
   not consent(R2,facebook);
situation1(R2,linkedin) when
   typeOf(R1,phone), typeOf(R2,data),
```

[8] We adopt the naming convention of an 'i' prefix (where this is unambigious) to denote an institutional event.

```
8     has(R1,R2), has(R1,linkedin_app),
9     not consent(R2,linkedin);
```

An example of this situation arises in S_1 and S_2 of Figure 4, after Rodger's data, for which linkedin has no consent, reaches Mark's phone, which has the linkedin app and likewise for Andrew's data in respect of facebook. In order to illustrate a possible modelling choice, the synchronization action here has been specified not to transfer data from the phone to the platform without the necessary consent:

```
1  sync(R1,R2) generates iSync(R1,R2);
2  iSync(R1,R2) initiates has(R2,R3) if
3     typeOf(R1,phone), typeOf(R2,platform),
4     has(R1,R3),typeOf(R3,data),
5     consent(R3,R2);
```

It can be seen in S_2 of Figure 4 that although there is a synchronization operation between Mark's phone and facebook, Mark's and Rodger's data is transferred to facebook, while Andrew's data is not. The use of a sticky policy, which is associated with the data, rather than with a particular device or platform, means that as long as a particular device observes the policy, the data subject's requirements can be met. However, if a particular device does not observe the `consent` policy, the data may be transferred anyway, in which case, detecting that the data is `vulnerable` comes too late. Consequently, as a further illustration of what might be possible or desirable, the model also recognizes when a friend location presents a risk, in advance of `send`ing data to that location:

```
1  warning(R,P) when situation2(R,P);
2  situation2(R1,facebook) when
3     typeOf(R1,data), has(R2,R1),
4     friend(R2,R3), has(R3,facebook_app),
5     not consent(R1,facebook);
6  situation2(R1,linkedin) when
7     typeOf(R1,data), has(R2,R1),
8     friend(R2,R3), has(R3,linkedin_app),
9     not consent(R1,linkedin);
```

An example of this circumstance occurs in the initial configuration of the trace in Figure 4, in which both Andrew's and Rodger's data are subject to warnings about facebook and linkedin, because Mark's phone has both SNP apps.

(c) The third circumstance the model considers is the case of sharing of data between platforms; that is, another kind of third-party action that can result in the transfer of data to a location which the suject has not authorized and/or is contrary to the subject's expressed policy. A further challenge is that the subject may be unaware of the identity of the organization that has gained access to the data and so a facility for a specific prohibition is of no help, suggesting a need also for quantified exclusion policy schema. The share action modelled here takes no account of any (sticky) policies:

```
1  share(R1,R2) generates iShare(R1,R2);
2  iShare(R1,R2) initiates has(R2,R3) if
3     typeOf(R1,platform), typeOf(R2,platform),
4     has(R1,R3),typeOf(R3,data);
```

Thus, even if some data is held with consent on one platform, there is the risk that it may be shared with another platform for which consent has not been given. This is reflected in `situation3`:

```
vulnerable(R,P) when situation3(R,P);
situation3(R1,R3) when
   has(R2,R1), typeOf(R1,data), typeOf(R2,platform),
   consent(R1,R2),
   typeOf(R3,platform), not consent(R1,R3);
```

and is illustrated in S_3 of Figure 4, where Rodger's data is on facebook and hence vulnerable to sharing with linkedin. It is conceivable that as a result of a further sharing, the data arrives on a platform that i. does observe the consent policy, and ii. consequently informs (due to the transparency policy) the subject of the receipt of the data, illustrating the kind of incomplete observability that is intrinsic to the problem. Such a situation is not modelled here for the sake of space and undue complication.

(d) Finally, there is the situation that it is sought to avoid, in which data resides on a platform without consent. This is recognized by the `compromised` condition:

```
compromised(R1,R2) when
   has(R2,R1), not consent(R1,R2),
   typeOf(R1,data), typeOf(R2,platform);
```

which translates as:

```
holdsat(compromised(R1,R2),I) :-
   holdsat(has(R2,R1),I),
   not
   holdsat(consent(R1,R2),I),
   holdsat(typeOf(R1,data),I),
   holdsat(typeOf(R2,platform),I),
   resource(R1),
   resource(R2),
   instant(I).
```

This occurs in S_4 of Figure 4, where following the sharing of data from facebook to linkedin, Rodger's data is now on linkedin. Clearly, this circumstance could be avoided by modelling the `share` operation to take account of consent, like the `sync` operation, but as noted above, this is a modelling decision, as part of the exploration process, to determine where vulnerabilities lie and which interventions may have the desired consequence of, say, no traces containing the `compromised` term.

JP: redrawn Fig 4 to improve legibility

The trace in Figure 4 results from evaluating a specific set of observations, starting from a specific configuration, for the purpose of the above narrative and to illustrate key features of the model, reflecting the situation shown in Figure 3. The same model can also be used to explore all possible traces, arising from all possible events in all possible orders [25], in order to carry out an exhaustive analysis. Such an approach implies a combinatorial state space explosion, but many event orders make no sense through either physical or logical impossibility and so can be discarded, so that in the case of this model that number of answer sets grows by a factor of $\approx$12 for each time step. Indeed, the designer may only be interested in traces containing certain kinds of

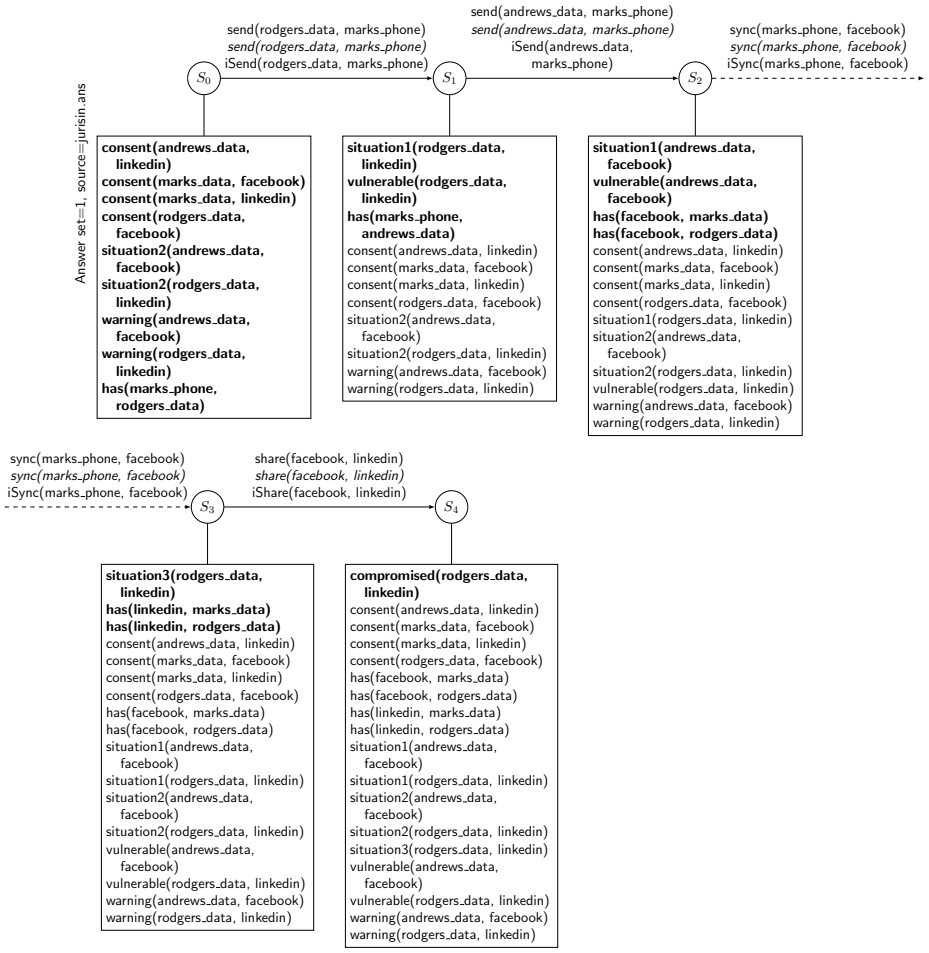

Fig. 4. The phone data model trace

situations (such as compromised) or determining where audit operations for subsequent compliance checking should be incorporated.

4.1 Query Size and Filtering

JP: new para

The space/time implications of the use of Answer Set Programming often raise concerns, but the technology is arguably a tangential aspect, since it is the problem domain that is actually the source of the complexity, while ASP is the means to tame it. In that respect, ASP is not different from any other tool for modelling and solving high-complexity problems, since the issue reduces then to which tool to use to answer the question. It is worth noting, the query that is used in conjunction with any model, in order to select the traces of interest, also has a significant (positive) impact on perfor-

mance by inhibiting the construction of unwanted results. As indicated in the preceding paragraph, exploratory usage is to generate all the answer sets that do not satisfy some generic exclusion criteria, such as:

```
:- observed(send(R,R),I),instant(I).
:- observed(sync(R,R),I),instant(I).
:- observed(share(R,R),I),instant(I).
```

where, for each (exogenous) event, traces that associate the event with the same entity are discarded. Likewise, we can discard any trace in which the target of a `send` (or other event) is a type name:

```
:- observed(send(R1,R2),I),instant(I),holdsat(subclassOf(R1,class),I).
:- observed(send(R1,R2),I),instant(I),holdsat(subclassOf(R2,class),I).
```

Examples such as these apply to many models, but there are also typically domain-specific exclusion criteria, such as:

```
:- observed(sync(R1,R2),I),instant(I),holdsat(typeOf(R1,R3),I),R3!=phone.
:- observed(sync(R1,R2),I),instant(I),holdsat(typeOf(R2,R4),I),R4!=platform.
```

which constrain the generated answer sets to those in which the `sync` event relates only a phone (`R1`) and a platform (`R2`).

The alternative is to generate only the answer sets that satisfy some criteria. For example, the trace shown in Figure 4, was the result of the inclusion criteria:

```
observed(send(rodgers_data,marks_phone),0).
observed(send(andrews_data,marks_phone),1).
observed(sync(marks_phone,facebook),2).
observed(share(facebook,linkedin),3).
#hide.
```

In addition to the means to control the number of time steps to explore, the combination of these two approaches provides the designer with a powerful filtering mechanism, in which constraints can be expressed both generically and in domain-specific terms. Thus, while absolute complexity inevitably remains in theory, useful solutions can frequently be obtained in practice.

5 Discussion

The approach we have outlined is a preliminary exploration of the use of answer set programming to generate possible outcomes for idealizations of complex scenarios as a means to develop a better understanding of, for example, potential risk, say to the security of data or system integrity. The exhaustive nature may bring a false sense of security however since, (i) the model evidence is only a 'proof' up to the fidelity of the model: any unmodelled (real world) feature may undermine the results, and (ii) of necessity, such a model can only explore a finite horizon, depending on time and compute resources available. These limitations do not render the approach pointless: complete information about a short time frame for as complete an understanding of the system as exists can be valuable in itself, as well as freeing designers to consider additional factors, with high confidence that at least certain interactions have been fully analysed. As such, the approach offers a means for damage limitation, rather than ensuring 100% solutions.

Lowering expectations about what can be achieved is necessary too in the context of the privacy and transparency scenario: complete privacy and complete transparency cannot be guaranteed – as illustrated by the observation in section 4 about data passing from a consent ignoring to a consent observing platform – but uncovering the points of interaction in a multi-agency environment where individuals are vulnerable and where organizations should provide transparency, is a necessary part of understanding the problem.

JP: new para

A legitimate concern, given the exhaustive nature of model construction and exploration is the space and time costs. Complexity is known to be NP in principle because that is an intrinsic property of the approach, but as we have demonstrated here (and in [25]), real-world constraints can reduce the actual search space to more manageable proportions for short finite periods, which capture time windows of interest.

The same model can also be used in a live setting [2] to meet some of the requirements sketched for the WIT model, simply by using short time horizons, typically just one event, to track the evolution of the institutional state against physical and software actions. In this circumstance, the complexity implications are less of a concern, since only a single time step is explored and so the exponential blow-up of longer time frames simple does not occur. Nevertheless, solution times do matter if (soft) real-time performance is required, but here the exhaustive nature can help, since the complexity of a model can be established from its structure and does not depend on the data input.

There are a number of significant challenges to tackle before it might be possible to deploy such systems. Among the more obvious shortcomings of the above, we would draw attention to the need for easy to use information [13], comprehensible information, the risk of deluge (for the scenario discussed in section 4) and the scale, scope and fidelity of the model. More substantive issues lie in a broader context, such as: (i) how to validate formal models of policy and of legislation against the event and state logs created by interactions (ii) generation of accessible explanations of what is happening and why, (iii) continuous validation, coupled with context awareness to be able to make appropriate decisions in respect of requesting the attention of the (data) subject or the auditor, or whichever other stakeholder is concerned, (iv) adapation to change in policy/legislation (v) policy capture (from policy makers) and policy extraction (from legacy policy represenations), and (vi) connection with semantic reasoning tools and suitable representations.

Acknowledgements The WIT model was developed by Pablo Noriega (IIIA, Barcelona) in conjunction with the first author, while the latter was on a short term visit to IIIA, supported by SINTELNET (FET Open Coordinated Action FP7-ICT-2009-C Project No. 286370).

References

1. C. Alchourrón and E. Bulygin. *Normative Systems*. Springer, 1971.
2. Tina Balke, Marina Vos, and Julian Padget. I-ABM: combining institutional frameworks and agent-based modelling for the design of enforcement policies. *Artificial Intelligence and Law*, pages 1–28, 2013.
3. Nelly Bencomo. Requirements for self-adaptation. In Ralf Lmmel, Joo Saraiva, and Joost Visser, editors, *Generative and Transformational Techniques in Software Engineering IV*,

volume 7680 of *Lecture Notes in Computer Science*, pages 271–296. Springer Berlin Heidelberg, 2013.
4. Liang Chen and Jason Crampton. Risk-aware role-based access control. In Catherine Meadows and M. Carmen Fernández Gago, editors, *STM*, volume 7170 of *Lecture Notes in Computer Science*, pages 140–156. Springer, 2011.
5. Liang Chen, Jason Crampton, Martin J. Kollingbaum, and Timothy J. Norman. Obligations in risk-aware access control. In Nora Cuppens-Boulahia, Philip Fong, Joaquín García-Alfaro, Stephen Marsh, and Jan-Philipp Steghöfer, editors, *PST*, pages 145–152. IEEE, 2012.
6. Yuan Cheng, Jaehong Park, and Ravi S. Sandhu. A user-to-user relationship-based access control model for online social networks. In Nora Cuppens-Boulahia, Frédéric Cuppens, and Joaquín García-Alfaro, editors, *DBSec*, volume 7371 of *Lecture Notes in Computer Science*, pages 8–24. Springer, 2012.
7. Owen Cliffe. *Specifying and Analysing Institutions in Multi-agent Systems Using Answer Set Programming*. PhD thesis, University of Bath, 2007.
8. Owen Cliffe, Marina De Vos, and Julian Padget. Answer set programming for representing and reasoning about virtual institutions. In Katsumi Inoue, Ken Satoh, and Francesca Toni, editors, *CLIMA VII*, volume 4371 of *Lecture Notes in Computer Science*, pages 60–79. Springer, 2006.
9. Owen Cliffe, Marina De Vos, and Julian Padget. Specifying and reasoning about multiple institutions. In Javier Vazquez-Salceda and Pablo Noriega, editors, *COIN 2006*, volume 4386 of *Lecture Notes in Computer Science*, pages 63–81. Springer, 2007. ISBN: 978-3-540-74457-3. Available via `http://dx.doi.org/10.1007/978-3-540-74459-7_5`.
10. Marina De Vos, Julian Padget, and Ken Satoh. Legal modelling and reasoning using institutions. In Takashi Onada, Daisuke Bekki, and Eric McCready, editors, *JSAI-isAI Workshops*, volume 6797 of *Lecture Notes in Computer Science*, pages 129–140. Springer, 2010.
11. Virginia Dignum and Julian Padget. Multiagent organizations. In Gerhard Weiss, editor, *Multiagent Systems*, pages 51–98. MIT Press, 2^{nd} edition, 2013. ISBN 978-0-262-01889-0.
12. Stephen Fickas and Martin S. Feather. Requirements monitoring in dynamic environments. In *RE*, pages 140–147. IEEE Computer Society, 1995.
13. A. Fung, M. Graham, and D. Weil. *Full Disclosure: The Perils and Promise of Transparency*. Cambridge University Press, 2007.
14. R Harré and P.F. Secord. *The Explanation of Social Behaviour*. Blackwells, 1972. ISBN 0 631 14220 7.
15. Hans Hedbom, Tobias Pulls, and Marit Hansen. Transparency tools. In Jan Camenisch, Simone Fischer-Hbner, and Kai Rannenberg, editors, *Privacy and Identity Management for Life*, pages 135–143. Springer Berlin Heidelberg, 2011.
16. Andrew Jones and Marek Sergot. A formal characterization of institutionalized power. *Logic Journal of the IGPL*, 4(3):427–446, 1996.
17. Günter Karjoth, Matthias Schunter, and Michael Waidner. Platform for enterprise privacy practices: privacy-enabled management of customer data. In *Proceedings of the 2nd international conference on Privacy enhancing technologies*, PET'02, pages 69–84, Berlin, Heidelberg, 2003. Springer-Verlag.
18. JeeHang Lee, Tingting Li, and Julian Padget. Towards polite virtual agents using social reasoning techniques. *Computer Animation and Virtual Worlds*, 24(3-4):335–343, 2013.
19. Yoram Moses and Moshe Tennenholtz. Artificial social systems part I: Basic principles. Technical Report CS90-12, Weizmann Institute, 1990.
20. Pablo Noriega, Amit K. Chopra, Nicoletta Fornara, Henrique Lopes Cardoso, and Munindar P. Singh. Regulated MAS: Social Perspective. In Giulia Andrighetto, Guido Governatori, Pablo Noriega, and Leendert W. N. van der Torre, editors, *Normative Multi-Agent Systems*, volume 4 of *Dagstuhl Follow-Ups*, pages 93–133. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2013.

21. Douglass C. North. *Institutions, Institutional Change and Economic Performance*. Cambridge University Press, 1991.
22. Kieron O'Hara. Transparent government, not transparent citizens: a report on privacy and transparency for the cabinet office. Technical report, September 2011.
23. Elinor Ostrom. *Governing the Commons. The Evolutions of Institutions for Collective Action.* Cambridge University Press, Cambridge., 1990.
24. Jaehong Park and Ravi Sandhu. The UCON$_{ABC}$ usage control model. *ACM Trans. Inf. Syst. Secur.*, 7(1):128–174, February 2004.
25. Wolter Pieters, Julian Padget, Francien Duchesne, Virginia Dignum, and Huib Aldewereld. Obligations to enforce prohibitions: On the adequacy of security policies. In *Proceedings of 6th International Conference on Security of Information in Networks*. ACM Press, 2013. Accepted for publication.
26. Bastin Tony Roy Savarimuthu, Stephen Cranefield, Maryam A. Purvis, and Martin K. Purvis. Obligation norm identification in agent societies. *Journal of Artificial Societies and Social Simulation*, 13(4):3, 2010.
27. Bastin Tony Roy Savarimuthu, Stephen Cranefield, Maryam A. Purvis, and Martin K. Purvis. Identifying prohibition norms in agent societies. *Artificial Intelligence and Law*, pages 1–46, 2012.
28. Peter Sawyer, Nelly Bencomo, Jon Whittle, Emmanuel Letier, and Anthony Finkelstein. Requirements-Aware Systems: A Research Agenda for RE for Self-adaptive Systems. In *RE*, pages 95–103. IEEE Computer Society, 2010.
29. John R. Searle. What is an institution? *Journal of Institutional Economics*, 1(01):1–22, 2005.
30. Thein Than Tun, Arosha K. Bandara, Blaine A. Price, Yijun Yu, Charles Haley, Inah Omoronyia, and Bashar Nuseibeh. Privacy arguments: Analysing selective disclosure requirements for mobile applications. *2012 20th IEEE International Requirements Engineering Conference (RE)*, 0:131–140, 2012.