A Methodology to Take Account of Diversity in Collective Adaptive Systems

Heather S. Packer and **Luc Moreau**[1]

Abstract. Collective Adaptive Systems (CASs) are comprised of a heterogeneous set of components often developed in a distributed manner. Their users are diverse with respect to their profiles, preferences, interests and goals, and hence, have different requirements. We propose a typology for the diversity of these components, users, and their requirements. We then present a methodology which provides steps to integrate features that record diversity to support accountability. The foundation of accountability is provided by provenance data, and a CAS vocabulary, these knowledge representation languages provide the core vocabulary that can be exploited by agents and services.

1 INTRODUCTION

Collective Adaptive Systems (CAS) are heterogeneous collections of autonomous task-oriented systems which contribute to a common goal, thus forming a collective system. The heterogeneous collections of systems means that CASs have diverse requirements because they have multiple stakeholders with different motivations, methods, tooling, profiles, and goals. There is also diversity in the way that each system processes the same data because of different perspectives and interpretations. It can be hard for participants to trust CASs because they are comprised of many systems which are often black boxes and strangers may be required to collaborate. Accountability in CASs enables its participants to build trust in the system and make informed decisions about other participants. In order to support the analysis of diversity in a CAS, it is important that their components adopt a standard model to express their properties and goals. CASs can also support diversity through the way that information is presented to different stakeholders, because they may require different types of information. For example, administrators might require statistics about usage, whereas a participant might require information about another participant to complete a task.

In CASs that rely on participants collaborating, reputation ratings and reviews are often used and can affect how members select or trust input from others. The algorithms and how participant use rating systems can vary greatly from CAS to CAS, therefore it can be hard to understand what ratings actual represent and mean to the community. Thus, it is important for a CAS's participants to understand how ratings are used and generated so that they can evaluate how to improve their rating or how much it should influence their selection process. It is also important that members can understand the potential utility of selecting others because strategically selecting members can maintain or improve their ratings. For example, some members may have high expectations and preserve high ratings for truly exceptional services, while others give high ratings more freely. Our focus is to provide end-users with an accountable CAS that instills trust in it and its diverse community.

Provenance is increasingly used for making systems accountable through exposing how information flows through a system and helping users to decide whether the resulting information can be trusted. The recent standard PROV [29] of the World Wide Web Consortium defines provenance as "a record that describes the people, institutions, entities, and activities involved in producing, influencing, or delivering a piece of data or a thing." PROV is a conceptual data model (PROV-DM [29]), which can be mapped and serialised to different technologies.

In order to provide accountability detailing diversity in CASs, we identify a typology for diversity in CASs and present a methodology to accommodate it. We use PROV which takes a Linked Data approach and benefits from its principles, namely through the use of Uniform Resource Identifiers (URIs), and the use of URIs to denote types for identifying resources. Diversity is supported through URIs, where individual and collectives are resources with properties, and those elements and properties are typed. In this paper we contribute:

1. A typology identifying diversity in CASs;
2. The CAS Vocabulary, which provides types for individuals and collectives;
3. A methodology for authoring provenance in CASs;
4. An approach that allows for diversity in an accountable way.

The rest of the paper as organised as follows. Section 2 describes our typology for diversity in CASs. In Section 3, we present the architecture of the platform to which we add accountability to diversity. Then, in Section 4 we describe an example application using the previously presented architecture. In Section 5 we detail our methodology. Then, in Section 6 we introduce the CAS vocabulary. In Section 7 we describe the diversity in applications. Following that in Section 8 we present how we use provenance and the CAS vocabulary to represent collectives and agents with different roles. In Section 9, we describe how queries can support different a range of requirements. Then in Section 10 we describe the reputation system and how we use the provenance data to describe diversity and present that data to a diverse set of end users. In Section 11, we summarised the features presented in the paper that support the diversity identified in Section 2 and discuss privacy and accountability. In section 12 we present related work to provenance and accountability. Finally, in Section 13 we conclude.

2 A TYPOLOGY OF DIVERSITY IN CASs

CAS are inherently diverse due to their human peers, components, stakeholders and goals. In order to support this diversity, we first

[1] University of Southampton, UK, email: {hp3, l.moreau}@ecs.soton.ac.uk

identify possible diversity in CASs:

1. Diversity in participants;
   (a) Human participants:
      i. The members of a CASs aim to achieve a common goal. However, each person has their own attributes, preferences, and perspectives.
      ii. People may opt to form a collective, where they formally aim to achieve a collective goal regardless of their differences.
      iii. People may be placed into a collective with or without their knowledge by a CAS, based on certain attributes which may including their actions or roles within a CAS.
      iv. The developers and designers of the CAS have a different perspective and different goals to the users of a CAS. They may consume different types of data to the end-users.
   (b) System components in a CAS have different responsibilities and roles within a system. They can be developed and hosted on different stacks and servers.
   (c) There are other types of participants, such as hardware agents using the CASs which may or may not aline with the goals of a CAS.
2. Diversity in interest:
   (a) While the members of a CAS work together to achieve a common goal, they can desire different outcomes based on their role and perspective. In a ride sharing example, one user is a driver and the other is a commuter, the driver main aim is to reduce the cost of travel, while the commuter requires transport.
   (b) People may require different information from the CAS. For example, some require information to support decisions or analyse the CAS.
   (c) People may also want information to be presented in different ways.
3. Diversity in roles and involvement in activities. While a community that uses a CAS might have common goals, the members may play different roles to achieve those goals. There is also diversity in the roles of data ownership, data stewardship, and data attribution.

Furthermore, these facets of diversity may change over time. This temporal dimension may affect the algorithm or components used within the CAS, interests may evolve over time, or the role of a CAS might change. This evolution may be unforeseen during design time, and thus, the design should cater for these evolving facets.

3 ARCHITECTURAL OVERVIEW OF AN ACCOUNTABLE CAS

In this section, we present the SmartSociety platform to situate how we provision for the accountability of diversity in the rest of the paper. The platform supports multiple CAS applications. Concretely, the core components of the platform are:

Peer Manager - This component manages the profiles of the platforms end-users. It is also an authentication service.
Application - An application consists of a group of components working together to support a common goal.
Component - Components in an application serve different purposes and may be developed by different developers.
Orchestration Manager - Handles the sequence of processes run by the components in an application.
Mobile Application - Mobile applications can be developed to allow end-users to interact with an application via the REST API.
Reputation Service - The reputation service manages feedback reports and generates reputation ratings for end-users.
Provenance Service - Stores provenance documents generated by the mobile applications, applications, reputation service, and orchestration manager.

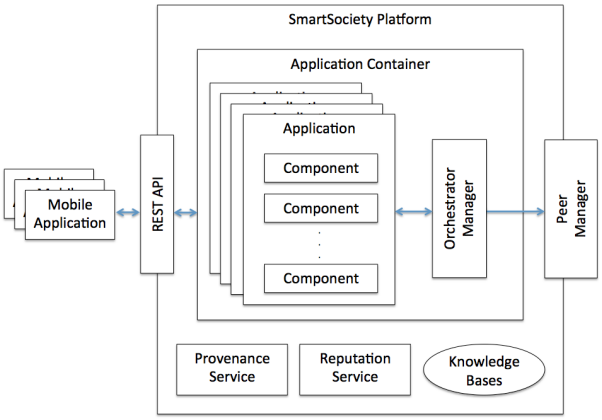

Figure 1. An overview of the architecture

4 RIDE SHARE

SmartShare is a car pooling application that allows drivers and commuters to offer and request rides. Ride offers and requests include details about required travels, timing, locations, capacity, prices, and other details relevant to car sharing. Specifically, this application is comprised of three core components, a Mobile Application, Orchestrator Manager and Reputation Service (see Figure 2). The application's orchestrator requests for a set of potential rides, which consists of a driver and commuters, from the Matcher. These potential rides are then agreed or rejected by its participants, this is handled by the Negotiator. Once a ride has been fulfilled, the drivers and commuters can leave each other feedback. This application is designed to be used in a diverse community, where its users range from office workers to tourists. The size and diversity of the community enables the application to be populated with many ride options, however, this diversity can cause problems in the application's adoption. Potential participants might be concerned with their safety with sharing rides with strangers from a diverse background.

SmartShare is provenance-enabled, capturing the provenance of any user decision, matching or rating managed by the system. The components in the architecture that record provenance are shaded in Figure 2. Specifically, the SmartShare application captures 10 processes that occur when:

1. A user logs into the mobile application;
2. A user changes a page on the mobile application;

3. The mobile application requests a resource from another service;
4. The mobile application submits a ride request to the orchestrator;
5. A composition of a ride is made by the orchestrator;
6. A ride is agreed on;
7. A ride is disagreed on;
8. A ride is agreed on all by all parties involved;
9. A reputation is generated;
10. A request is made via the reputation's API.

The provenance records a user's actions and how outcomes are generated, such the classification of a star or reputation ratings. The purpose of capturing provenance in SmartShare is to make the application accountable, in particular, by providing explanations about all decisions made. Its is required to be transparent and accountable to both the developers, and its end-users.

5 A METHODOLOGY FOR ACCOUNTABILITY IN CASs

In this section, we present our methodology which provide steps to integrate features that record diversity to support accountability. The steps in the methodology are iterated over to provide refinement through observations and changing requirements. The individual steps in this methodology need to be performed in a domain-specific way for each individual CAS. At a general level, our methodology consists of the following steps (see Figure 3):

1. CAS Vocabulary Development - Build a vocabulary to define types for agents, entities, and activities specific to a CAS. This step defines the conceptual space within which the system will operate, and specifies what processes fall within the boundaries of the system.
2. Component Design and Implementation - Design the interaction model(s) underlying the social computations that should be supported by the CAS. In this step, the protocols that will govern interactions are specified in terms of communication between interacting peers, the control flow of the collective coordination procedure, data access and synchronisation through shared state.
   During the second iteration of this step, the logging of the values for the variables defined in the template are generated.
3. Design Provenance Templates - Map the vocabulary and design the provenance that the system will capture. Provenance can capture the creation, modification and use of entities within the CAS. In order to model the provenance, we use PROV-Template[23], which is a declarative approach where the design of the provenance's semantics are separated from the logging of values recorded in the provenance, in templates and bindings, respectively. Provenance documents are generated by an expansion algorithm, which combines a template with a set of bindings. The granularity of the semantics captured in the provenance models may be modified through iterations of the methodology, to support different stakeholders requirements.
4. Define Requirements for Analysis and Social Computations - Define querying and summarisation functionalities for different stakeholders. These will produce the analysis facilities the system provides to human and machine peers for its analysis, and have to be adapted to the needs of the stakeholders involved, as well as to their interpretations (e.g. summaries for the platform operator might be different than for end users). The results from the queries can support approaches to express the information in provenance to end-users. In Figure 3, we show that the queries generated by this step use the PROV documents generated by PROV-Template's expansion algorithm, and the results are used to present information from the queries in different ways.

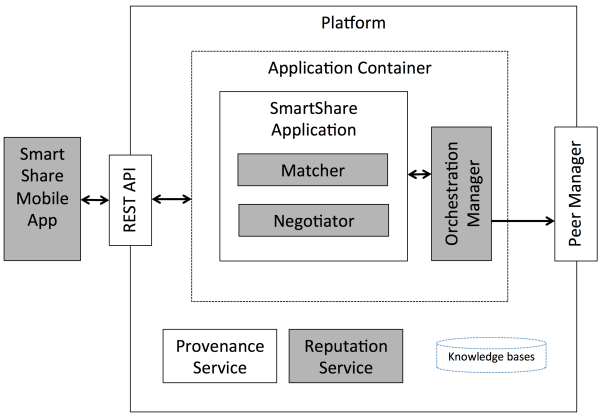

Figure 2. SmartShare Architecture

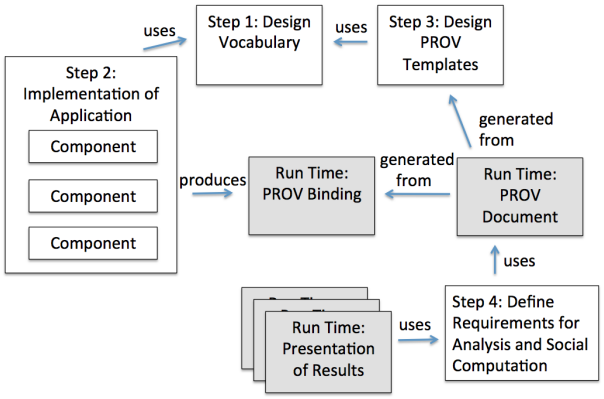

Figure 3. General Methodology

6 DIVERSITY IN VOCABULARY

In order for CASs to allow for diversity in their accounting, we require a way to differentiate between different facets of diversity identified in Section 2. Hence, in this section, we provide a vocabulary that defines types that can be used to differentiate between these facets. The diversity in a CAS may differ depending on its purpose and participants, therefore, we have designed an upper level vocabulary, which is designed to be extended to support CASs.

The core CAS vocabulary defines a hierarchy of sub-types branching from three key elements, agents, entities, and activities (see Figure 4). Specifically, the vocabulary focuses on describing three components: (1) agents within CASs, including users, peers, and collectives; (2) activities; and, (3) entities describing: outcomes of activities; and attributes of agents including preferences, capabilities, and goals.

Concretely, the vocabulary supports diversity:

1. In participants by defining (i) **cas:Peer** for CASs components

(ii) **cas:User** of the CASs (iii) **cas:Agent** are non-human agents that use CASs and (iv) **cas:Collective** that can be composed of **cas:Peer**, **cas:User** and **cas:Agent**.
2. In interests by defining **cas:Interest**.
3. In roles by defining **cas:Role** and **cas:Capability**.

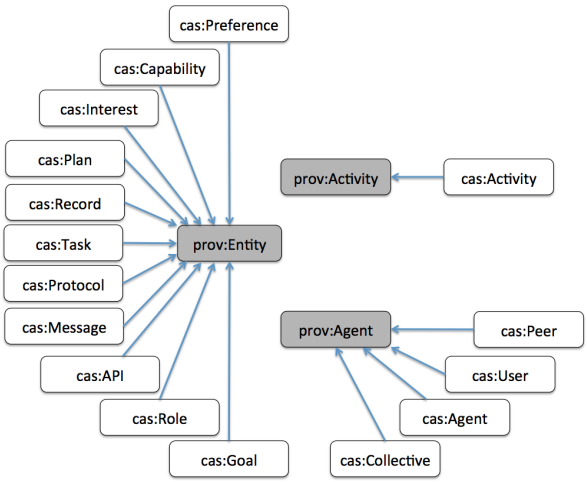

Figure 4. CAS Vocabulary

In the ride share example, the **cas:Peer** can be used to describe the orchestration manger, matcher, negotiator and reputation service. These programs play a distinct role with in the system, hence we can extend the vocabulary to include them using the following terms **cas:OrchestratorManager**, **cas:MatcherManager**, **cas:NegotiatorManager** and **cas:ReputationService**. The users play two distinct roles, driver and rider, we use the **cas:Role** to define **cas:Driver** and **cas:Commuter** (see Figure 5).

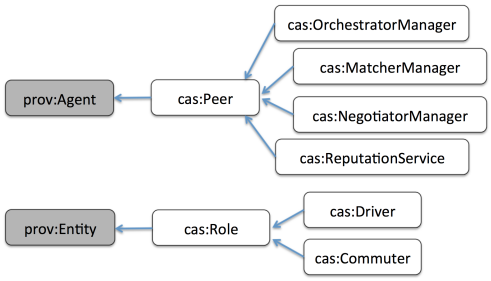

Figure 5. CAS Vocabulary Extension of Roles and Peer

7 DIVERSITY IN APPLICATION

In order to account for diversity in an application its resources are required to be described with URIs and typed with the CAS vocabulary. The URIs provide links to resources that were created, modified and used in the system, and provide context about the state of **cas:User**, **cas:Agent**, **cas:Peer** and **cas:Collective**.

An application's purpose may change during its use, therefore they need to allow for new diverse facets to be supported. For example, this supports the merging of two CASs or new types of participants to be added to the application. These changes will be required to be reflected in the application's vocabulary.

The SmartSociety platform, presented in Section 3, caters for a wide range of applications. The applications are contained in the Application Container, where each application's components are managed by its own orchestrator. The applications can interact with the reputation service and store provenance documents in the provenance service.

8 DIVERSITY IN PROVENANCE

Provenance templates are used to describe patterns to be captured by a system. In order to model the diversity, we describe how the features of PROV and the CAS Vocabulary can be used. PROV is a recent set of recommendations of the W3C for representing provenance on the web (see Figure 6). PROV is a conceptual data model (PROV-DM [29]), which can be mapped and serialized to different technologies. There is an OWL2 ontology for PROV (PROV-O [20]), allowing mapping to RDF, an XML schema for provenance [15], and a textual representation for PROV (PROV-N [30]). Provenance templates allow for diverse levels of logging.

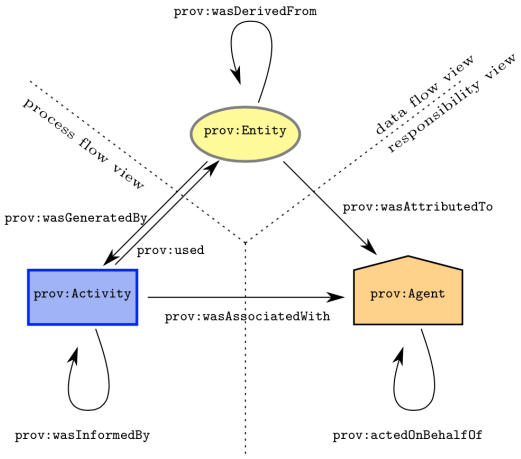

Figure 6. Three Different Views of the Core of PROV. The figure adopts the PROV layout conventions: an entity is represented by a yellow ellipsis, an activity by a blue rectangle, and an agent by an orange pentagon. We note here that the diagram is a "class diagram" illustrating the classes that occur as domain and range of properties. Taken from[27].

A user's diversity can be expressing using entities that are attributed to a prov:Agent. Specifically, **cas:Profile**, **cas:Preference**, **cas:Capability** can be used to type these entities (see Figure 7).

Collectives can be formed using **prov:Agent**s of type **cas:Collective** (see Figure 8). Collectives can also be organised into groups by their attributes. For example, Figure 9 shows an example where Alice and Bob are in a collective based on their capability of being able to drive.

A component or participant may play different roles within a CAS. A role can be express using the **cas:Role** and its type with **prov:type**. For example, Figures 10, 11 and 12 shows specialisations of a user with the roles of Driver, Commuter, and as a Commuter in a collective, respectively.

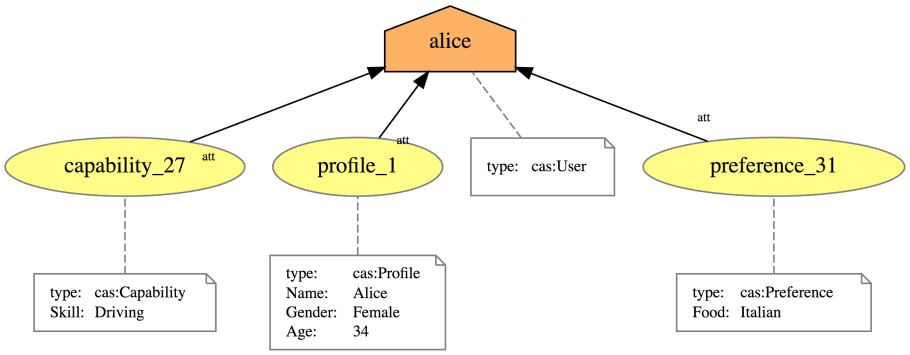

Figure 7. Alice's diversity show in her profile, preferences and capabilities

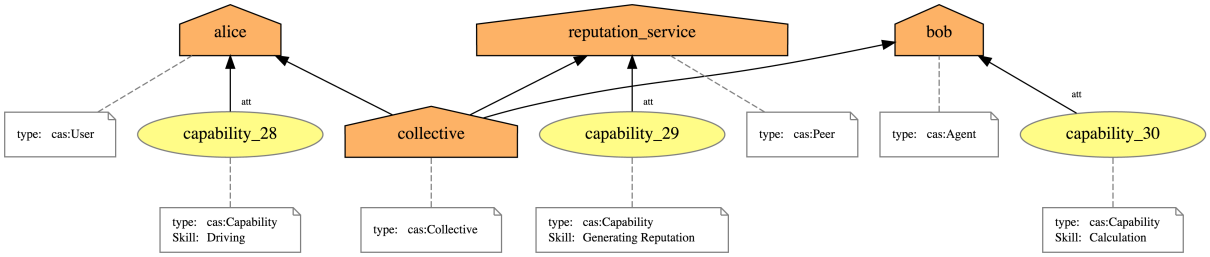

Figure 8. A collective with three different types of agent

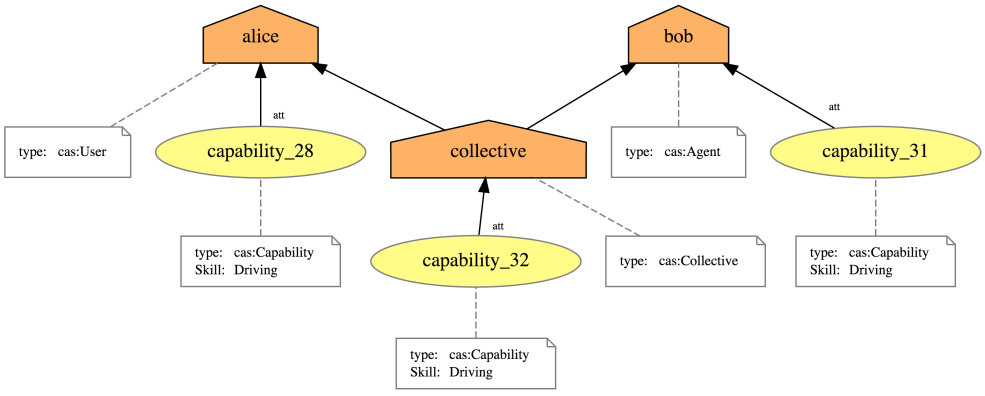

Figure 9. An ad-hoc collective connected by their capability to drive

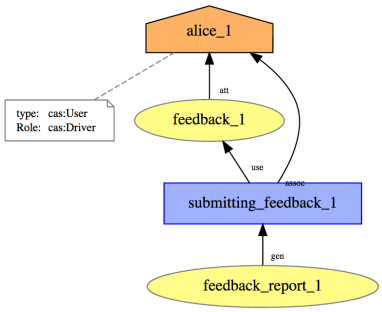

Figure 10. Alice in the role driver

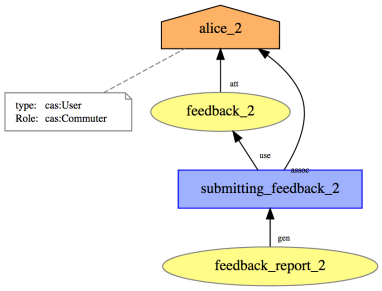

Figure 11. Alice in the role commuter

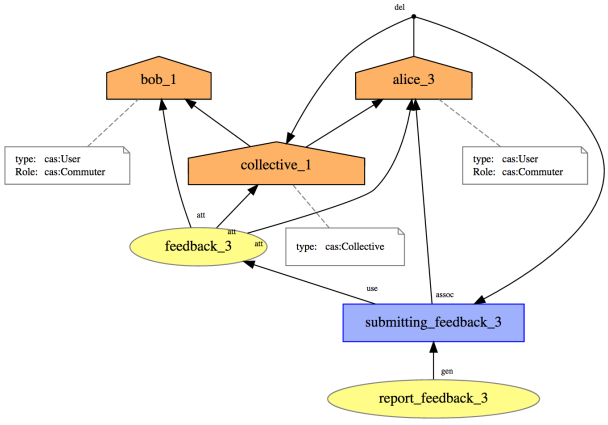

Figure 12. Alice in the role commuter submitting a feedback report, on behalf of a collective whose members were all connected to the submitted feedback report.

In Figure 13, the specialisations of Alice are linked to an Alice's generalisation, the model also captures state changes of Alice using derivations. Modelling an agent in this way enables them to act in clearly defined roles, and using PROV to define generalisations of an agent provides a clear hierarchical structure.

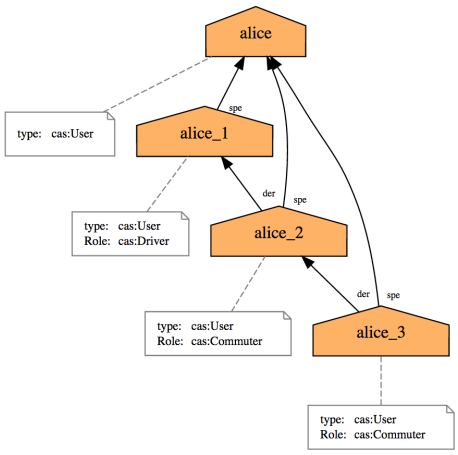

Figure 13. Provenance of linking of the specialisations of Alice to her generalisation

9 DIVERSITY IN QUERIES

Provenance graphs can be queried to support a diverse range of interests from:

End-users - Who may require specific information about themselves, others in their collective, or how a certain entity was generated so they can make informed decisions. End-users may find it easier to trust or act in transactions or collectives, if they know that they are collaborating with like minded people or in a collective with a particular range of skills.

Developers - Who may require statistics that provide them information about how end-users use the CAS, so that they can make improvements to popular features or stop supporting those that are unused.

Administrators - Who may require summative information about usage statistics and verify that entities are created following the CASs protocols.

Software Components - May require an aggregation to provide an input to a function so that the CAS can adapt to how it is being used.

Provenance expressed an RDF syntax can be queried with SPARQL. For example, the following SPARQL query returns all the users that are associated with collectives:

```
SELECT ?user WHERE {
  ?user prov:hadMember ?collective
  ?collective prov:type cas:Collective
}
```

The following SPARQL query returns all collectives that contain one or more drivers:

```
SELECT ?collective WHERE {
  ?user prov:hadMember ?collective
  ?collective prov:type cas:Collective
  ?user cas:Role driver
}
```

These queries can be used to support administrators or software components. Using this linked data approach supports different perspectives on the data.

10 DIVERSITY IN EXPLANATIONS

It is often necessary for CASs and their users to make decisions based on provenance data. Therefore, it is important to communicate that data accessibly to both machines to support adaptability in their algorithms and human users to facilitate transparency and accountability. Provenance data is machine readable, the largest challenge is communicating data with humans in a diverse way. It is possible to use graphical approaches for this, but, in many cases, it is more natural or appropriate to communicate this data either textually or verbally. We identified in [32] that the largest challenges related to utilising the structure and elements in provenance data were:

1. Identifying and presenting interesting facts to support a particular use case. For example, a user can view an explanation about someone else to aid in their decision to share a ride with them, a narrative can provide evidence of reliability from the provenance based on features such as the number of feedback reports left by a user, or the number of times this user has interacted with the application;
2. How to describe PROV elements without referring to long complicated URIs, while providing meaningful explanations. Long URIs break up the fluidity of sentences making them hard for humans to parse.

In order to mitigate these issues, we have created an approach to convert provenance data into a linear, textual form. In more detail, the steps of this narrative approach are:

1. Identify information is relevant to the target audiences. This step should involve an exploratory study involving potential users from a diverse set of backgrounds, profiles, and roles, which investigates the information users require to support their decision making;
2. Author queries to extract the identified information. These queries can extract direct values or aggregations from the data. For example, provenance can be queried for an instance of a particular type or provenance can be quiet for the number of instances of a particular type;
3. Author sentences templates for the target audiences appropriate for those identified in Step 1. in both first and third-person perspectives (see Table 1 for examples of sentence template). The sentence templates are authored in HTML so that they can benefit from hyperlinks. The sentence templates utilise the CAS types to refer to a resource so that we can reduce the number of URIs in the narrative, however, these URIs are preserved in the narrative through hyperlinks;
4. Execute the queries and identify which templates can be fulfilled based on the queries' results. Using those identified queries, replace the variables with the values from the queries.

These descriptions can be embedded into CASs to show in a transparent manner how resources are used and generated by the system. They can be used to describe user behaviour, which helps increase users' awareness of others and their actions, thus supporting accountability.

11 DISCUSSION

Our methodology aims to aid in the design of CASs, models of diversity, support diverse analysis requirements and provide accountability to continue to encourage a diverse community. Recording diversity in CASs means that it can be analysed throughout its components and participants. The transparency of diversity in CASs can enable developers to formulate approaches to support diversity. For example, developers could develop an incentive for participants to complete tasks with others that have different skills to them. Transparency of diversity also promotes trust within its community. For example, participants who share accommodation may want to share with others that have similar preferences and hobbies, people tend to trust others how are similar to themselves.

In the following Table 2, we discuss how we address the diversity typology that we identified in Section 2. Temporal aspects of diversity mentioned in Section 2 have been catered for by using the **prov:wasDerivedBy** relationship typed with **prov:Revision**. This enables the state changes to be modelled, and can show how profiles, preference, capabilities, goals and roles evolve (see Figure 13).

While accountability affords many benefits, it may, however, lead to breaches in privacy. Diversity may be expressed in PROV using types and properties, which are regarded as private data. Semantic inference may lead to exposing private diversity information. For example, the following statements expose X and Y's sexual preferences.

```
X and Y are involved in a marriage activity
    -> X and Y may or may not be of the same gender

U and V are involved in a civil partnership
    -> U and V are same sex couple
```

It is, therefore, important that a Privacy Impact Assessment (PIA) should be completed during each iteration of designing provenance templates in our methodology (Step 3 in Section 5). Specifically, PIA is a tool that you can use to identify and reduce the privacy risks. A PIA can reduce the risks of harm to individuals through the misuse of their personal information. It can also help you to design more efficient and effective processes for handling personal data.

In order to show the diversity in the SmartShare application, we describe how diversity is allowed for in the application in Table 3.

12 RELATED WORK

The diversity of people in online systems has long been recognised [37, 7, 33]. There have been numerous studies evaluating cultural differences in online systems [4, 21, 1, 10], which identify that diversity plays a big in the outcome of these systems. These types of evaluations often heavy rely on user studies and provide no standard models which can be used to provide comparisons between different systems.

There has been a large body of work advocating accountability in distributed systems [19, 2, 34], which handle a diverse set of system component. The work presented in [2] describes the role of accountability in distributed systems. They identify that accountability makes it "possible to tolerate, detect, isolate, discourage, and remove misbehaving components". In CASs accountability can play

Description	First Person	Second Person
Collective	You and {list_of_users} took part in a ride as {role}.	{list_of_users} took part in a ride as {role}.
Reputation Report	You have an average overall rating of {average_rating} from {no_feedback_reports} feedback reports left by {no_authors} authors. Out of the {total_feedback_reports} feedback reports written by you, only {no_feedback_reports} were used to generate your rating. The feedback reports used to generate your rating were authored in the last {no_days} day/s.	This user has an average overall rating of {average_rating} from {no_feedback_reports} feedback reports left by {no_authors} authors. Out of the {total_feedback_reports} feedback reports written about them, only {no_feedback_reports} were used to generate the rating. The feedback reports used to generate that rating were authored in the last {no_days} day/s.
Users Behaviour	You have left an average feedback of {average_feedback}, and have written {no_authored_reports} feedback reports. You have left feedback for {no_unique_users} different users, and it agrees with {agreement_percentage}% of the other raters. Your feedback that does not agree with other raters was {disagreement_percentage}% higher.	This user left an average feedback of {average_feedback}, and has written {no_authored_reports} feedback reports. They have left feedback for {no_unique_users} different users, and it agrees with {agreement_percentage}% of the other raters. The feedback that does not agree with other raters was {disagreement_percentage}% higher.

Table 1. Sentence templates supporting both first and second person perspective, elements surrounded by {} are variables.

the same role, where components and users can detect misbehaving components or users.

Provenance can be used to describe the flow of information and human participation in activities. Applications that record provenance and provenance use cases are well documented [3, 24, 25, 13]. Moreover, the use cases include support for: making social computations accountable and transparent [36, 31]; determining whether data or users can be trusted [16]; and ensuring reproducibility [26] of computations; auditability and accountability [36]; deriving trust and classification [17]; asserting attribution and generating acknowledgements [27]; and traceability [8]. To enable such a powerful functionality, however, one needs to adapt or write applications, so that they generate provenance information, which can then be exploited to offer new benefits to their users. Provenance can be generated during runtime [11, 14, 28], compile time [6, 5], and reconstructed retrospectively [22, 9].

Previously, Semantic Web technologies have been used to generate narratives [35, 18, 12]. In more detail, Tuffield et al. [35] and Jewell et al. [18] describe the OntoMedia ontology, which supports the generation of narratives. Tuffield et al. [35] discuss approaches to generate narratives from a vocabulary, the approaches included are based on character, plot and user modelling. Jewell et al. [18] describes how OntoMedia is used to annotate the vast collection of heterogeneous media. Geurts et al. [12] use ontological domain knowledge to select and organise a narrative discourse on a topic of interest to a user.

13 CONCLUDING REMARKS

In this paper, we present a typology for diversity in CASs and a methodology to aid in the design of CASs, models of diversity, support diverse analysis requirements and provide accountability to continue to encourage a diverse community. Supporting diverse analysis requirements promotes trust and familiarity in the CAS and its participants. This transparency allows its participants to view how components and others behave. Thus, it enables its participants more information when to support their choice to contributing to a transaction and or joining collectives. The methodology presented in this paper draws on linked data principles to provide the basis of an information model that is diversity aware and supports reuse. PROV and the CAS vocabulary allow the actions of CASs peers to be modelled, this model can be exploited by other services to support end-users or adaptive algorithms.

The narrative approach is one such example of how to convey and support diversity, by enabling provenance information about reputation to be consumed easily by humans with different perspectives. We have planned an evaluation, to evaluate explanations from the provenance generated by the reputation service that will enable users to understand (1) how their reputation is generated, which takes into account the decay of feedback reports; (2) recommendations of which subject to choose, which are motivated by whether a subject routinely leaves feedback and whether they rate highly, which contrasts to using just a reputation rating to support decision-making; and (3) how they are perceived by others, which aims to increase their awareness that their actions within an CAS have consequences.

ACKNOWLEDGEMENTS

The research leading to these results has received partially funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n. 600854 Smart Society: hybrid and diversity-aware collective adaptive systems: where people meet machines to build smarter societies http://www.smart-society-project.eu/.

REFERENCES

[1] Alexandre Ardichvili, Martin Maurer, Wei Li, Tim Wentling, and Reed Stuedemann, 'Cultural influences on knowledge sharing throughtr online communities of practice', *Journal of knowledge management*, **10**(1), 94–107, (2006).

[2] Elisa Bertino, Wonjun Lee, Anna C Squicciarini, and Bhavani Thuraisingham, 'End-to-end accountability in grid computing systems for coalition information sharing', in *Proceedings of the 4th annual workshop on Cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead*, p. 29. ACM, (2008).

[3] Rajendra Bose and James Frew, 'Lineage retrieval for scientific data processing: A survey', *ACM Computing Surveys*, **37**(1), 1–28, (March 2005).

Diversity	Support
1. Diversity in Participants	The participants in a CAS are typed with labels stemming from prov:Agent
(a) Human Participants	In the CAS Vocabulary human participants are described as **cas:User**. A **cas:User** can have attributes of **cas:Profile**, **cas:Preference** and **cas:Capability** (see Figure 7). In more detail: **cas:Profile** resources provide additional information about a user which may include details about their gender, age or allergies. **cas:Preference** resouces describe a users preferences, which may include a user's homepage, contact hours or type of food. **cas:Capability** resources describe a user's skill, which may include recognising star constellations, driving, or taking photographs. These types are assigned to PROV elements during the Step 3 of our methodology presented in Section 5. Any additional types required to define an agent's profile, preferences and capabilities for a specific purpose would be defined in Step 1 (see Section 5).
i. Individual Users	See the description for 1. (a)
ii. Collectives	An agent of type **cas:Collective** can act on behalf of others with the type **cas:User**, **cas:Peer**, **cas:Agent** (see Figure 8). Each of these agents can be resources of type **cas:Profile**, **cas:Preference** and **cas:Capability**. These types are assigned during the Step 3 of our methodology (see Section 5).
iii. Ad-Hoc Collectives	In order to differentiate between collectives that have been created by their users the collective can be attributed to entities of type **cas:Profile**, **cas:Preference** and or **cas:Capability** which describe the connection between the entities (see Figure 9). These types are assigned during the Step 3 of our methodology (see Section 5).
iv. Developers and Designers	Can be described by using the type **cas:User**. Their roles, profiles and capabilities can be typed with **cas:Role**, **cas:Profile** and **cas:Capability**, respectively. These types are assigned to PROV elements during the Step 3 of our methodology (see Section 5).
(b) System Components	Have type **cas:Peer**, which can be described by resources of type **cas:Profile** and **cas:Capability**. These types are assigned during the Step 3 of our methodology (see Section 5).
(c) Other	Have type **cas:Agent**, which can be described by resource of type **cas:Profile** and **cas:Capability**. These types are assigned during the Step 3 of our methodology (see Section 5).
2. Diversity in Interest	Can be described in the provenance captured from the CAS, this diversity can be queried using SPARQL, and the results from queries can be presented in different ways.
(a) Different Goals	Can be captured in the provenance by using the type **cas:Goal**. This type is assigned to PROV elements during the Step 3 of our methodology (see Section 5).
(b) View different Information	A wide range of queries can be authored to retrieve information from the provenance. These queries are designed during the Step 4 of our methodology (see Section 5).
(c) Alternative Presentation	The information from queries can be presented in different ways, see Section 10 for an example of sentence templates which can use a first person or third person perspective to describe the data. These sentence templates would be designed during the Step 4 of our methodology (see Section 5).
3. Diversity in Roles	Can be captured in the provenance by using the type **cas:Role**. This type is assigned to PROV elements during the Step 3 of our methodology (see Section 5).

Table 2. Table describing how our approach supports the diversity typology in Section 2.

[4] Patrick YK Chau, Melissa Cole, Anne P Massey, Mitzi Montoya-Weiss, and Robert M O'Keefe, 'Cultural differences in the online behavior of consumers', *Communications of the ACM*, **45**(10), 138–143, (2002).

[5] James Cheney, 'Program slicing and data provenance', *IEEE Data Engineering Bulletin*, 22–28, (December 2007).

[6] James Cheney, Amal Ahmed, and Umut A. Acar, 'Provenance as dependency analysis', *Mathematical Structures in Computer Science*, **21**(6), 1301–1337, (2011).

[7] BA Collis and Elka Remmers, 'The www in education: issues related to cross-cultural communication and interaction', (1997).

[8] Francisco Curbera, Yurdaer Doganata, Axel Martens, Nirmal K. Mukhi, and Aleksander Slominski, 'Business provenance – a technology to increase traceability of end-to-end operations', in *OTM 2008 Confederated International Conferences*, eds., Robert Meersman and Zahir Tari, pp. 100–119. Springer, (2008).

[9] Tom De Nies, Io Taxidou, Anastasia Dimou, Ruben Verborgh, Peter M. Fischer, Erik Mannens, and Rik Van de Walle, 'Towards multi-level provenance reconstruction of information diffusion on social media', in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pp. 1823–1826, New York, NY, USA, (2015). ACM.

[10] Casey Dugan, Werner Geyer, Michael Muller, Joan DiMicco, Beth Brownholtz, and David R Millen, 'It's all'about you': diversity in online profiles', in *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pp. 703–706. ACM, (2008).

[11] James Frew, Dominic Metzger, and Peter Slaughter, 'Automatic capture and reconstruction of computational provenance', *Concurrency and Computation: Practice and Experience*, **20**(5), 485–496, (2008).

[12] Joost Geurts, Stefano Bocconi, Jacco Van Ossenbruggen, and Lynda Hardman, *Towards ontology-driven discourse: From semantic graphs*

Diversity	Support
1. Diversity in Participants	
(a) Human Participants	
i. Individual Users	SmartShare users are typed with **cas:User**, and a driver has the skill driver.
ii. Collectives	In SmartShare individuals can select to be in the same collective to share a ride after they have been matched because they share the same departure, destination and time. This is modelled in the provenance captured.
iii. Ad-Hoc Collectives	In SmartShare individuals are placed into ad-hoc collectives by the matching service based on their departure and destination locations and times. The matching service then proposes these matches to the matched users. This is modelled in the provenance captured.
iv. Developers and Designers	The SmartShare application has different components each component has a different team of developers.
(b) System Components	SmartShare is comprised of Orchestrator, Negotiation, Matcher, and Reputation services, and a mobile application. Each of these services have their own type **cas:Orchestrator**, **cas:NegotiationManager**, **cas:MatchingManager**, **cas:ReputationManager**, and **cas:MobileApplication**, respectively, these types stem from **cas:Peer**.
(c) Other	For now, there are no other types of participants. This could change if driverless cars were part of the system.
2. Diversity in Interest	
(a) Different Goals	The end-users have different goals, some want to save money on their commute while others want a transportation. This is represented in the provenance using the **cas:Goal** type. The developer for each component also have different goals, for example, the orchestrator aims to organises tasks as efficiency as possible, while the reputation services aims to generate reputation reports.
(b) View Different Information	User's of the system want to see which ride match their requirements and reputation information about each ride participant. This information is modelled in the provenance. Administrators want an overview of the activity on SmartShare for example how many rides have been completed, where the most rides have taken place, and how many new users there are. These figures can are obtained from the provenance.
(c) Alternative Presentation	Provenance information can be displayed as graphs to the developers and textual information using sentence templates to end-users. The reputation service generates text to describe how reputation is generated and statistics about how a user interacts with the reputation service.
3. Diversity in Roles	The end-users can have two types of role, **cas:Driver** and or **cas:Commuter**.

Table 3. Table describing how our approach supports the diversity typology in Section 2.

to multimedia presentations, Springer, 2003.

[13] Yolanda Gil, James Cheney, Paul Groth, Olaf Hartig, Simon Miles, Luc Moreau, and Paulo Pinheiro da Silva, 'Provenance xg final report', Technical report, World Wide Web Consortium, (2010).

[14] David A. Holland, Margo Seltzer, Uri Braun, and Kiran-Kumar Muniswamy-Reddy, 'Pass-ing the provenance challenge', *Concurrency and Computation: Practice and Experience*, **20**(5), (2008).

[15] Hook Hua, Curt Tilmes, Stephan Zednik (eds.), and Luc Moreau, 'PROV-XML: The PROV XML Schema', W3C Working Group Note NOTE-prov-xml-20130430, World Wide Web Consortium, (April 2013).

[16] Trung Dong Huynh, *Trust and reputation in open multi-agent systems*, Ph.D. dissertation, University of Southampton, 2006.

[17] Trung Dong Huynh, Mark Ebden, Matteo Venanzi, Sarvapali Ramchurn, Stephen Roberts, and Luc Moreau, 'Interpretation of crowdsourced activities using provenance network analysis', in *Conference on Human Computation and Crowdsourcing (HCOMP'13)*, (November 2013).

[18] Michael O Jewell, K Faith Lawrence, Mischa M Tuffield, Adam Prugel-Bennett, David E Millard, Mark S Nixon, Nigel R Shadbolt, et al., 'Ontomedia: An ontology for the representation of heterogeneous media', in *In Proceeding of SIGIR workshop on Mutlimedia Information Retrieval*. ACM SIGIR, (2005).

[19] Ryan KL Ko, Peter Jagadpramana, Miranda Mowbray, Siani Pearson, Markus Kirchberg, Qianhui Liang, and Bu Sung Lee, 'Trustcloud: A framework for accountability and trust in cloud computing', in *Services (SERVICES), 2011 IEEE World Congress on*, pp. 584–588. IEEE, (2011).

[20] Timothy Lebo, Satya Sahoo, Deborah McGuinness (eds.), Khalid Behajjame, James Cheney, David Corsar, Daniel Garijo, Stian Soiland-Reyes, Stephan Zednik, and Jun Zhao, 'PROV-O: The PROV Ontology', W3C Recommendation REC-prov-o-20130430, World Wide Web Consortium, (October 2013).

[21] Doo H Lim, 'Cross cultural differences in online learning motivation', *Educational Media International*, **41**(2), 163–175, (2004).

[22] Sara Magliacane, 'Reconstructing provenance', in *The Semantic Web — ISWC 2012*, volume 7650 of *Lecture Notes in Computer Science*, pp. 399–406. Springer, (2012).

[23] Danius Michaelides, Trung Dong Huynh, and Luc Moreau. Provtemplate: A template system for prov documents, June 2014. Technical Note.

[24] Simon Miles, Paul Groth, Miguel Branco, and Luc Moreau, 'The requirements of recording and using provenance in e-science experiments', *Journal of Grid Computing*, **5**(1), 1–25, (2007).

[25] Luc Moreau, 'The foundations for provenance on the web', *Foundations and Trends in Web Science*, **2**(2–3), 99–241, (November 2010).

[26] Luc Moreau, 'Provenance-based reproducibility in the semantic web', *Web Semantics: Science Services and Agents on the World Wide Web*, **9**(2), 202–221, (July 2011).

[27] Luc Moreau and Paul Groth, *Provenance: An Introduction to PROV*, Morgan and Claypool, September 2013.

[28] Luc Moreau and Paul Groth, 'Provenance of publications: A prov style for latex', in *Seventh USENIX Workshop on the Theory and Practice of Provenance (TAPP'15)*, Edinburgh, Scotland, (July 2015). USENIX.

[29] Luc Moreau, Paolo Missier (eds.), Khalid Belhajjame, Reza B'Far, James Cheney, Sam Coppens, Stephen Cresswell, Yolanda Gil, Paul Groth, Graham Klyne, Timothy Lebo, Jim McCusker, Simon Miles, James Myers, Satya Sahoo, and Curt Tilmes, 'PROV-DM: The PROV Data Model', W3C Recommendation REC-prov-dm-20130430, World Wide Web Consortium, (October 2013).

[30] Luc Moreau, Paolo Missier (eds.), James Cheney, and Stian Soiland-Reyes, 'PROV-N: The Provenance Notation', W3C Recommendation REC-prov-n-20130430, World Wide Web Consortium, (October 2013).

[31] Heather S Packer, Laura Drăgan, and Luc Moreau, 'An auditable reputation service for collective adaptive systems', in *Social Collective Intelligence*, 159–184, Springer, (2014).

[32] Heather S Packer and Luc Moreau, 'Generating narratives from provenance relationship chains', in *Proceedings of the 2015 Workshop on Narrative & Hypertext*, pp. 37–41. ACM, (2015).

[33] John P Robinson, Alan Neustadtl, and Meyer Kestnbaum, 'The online "diversity divide": Public opinion differences among internet users and nonusers', *IT & Society*, **1**(1), 284–302, (2002).

[34] Thomas Ruebsamen and Christoph Reich, 'Supporting cloud accountability by collecting evidence using audit agents', in *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, volume 1, pp. 185–190. IEEE, (2013).

[35] Mischa M Tuffield, Nigel R Shadbolt, and David E Millard, 'Narrative as a form of knowledge transfer: Narrative theory and semantics', in *In Proceedings of the First AKT DTA Colloquium*, (2005).

[36] Daniel J. Weitzner, Harold Abelson, Tim Berners-Lee, Joan Feigenbaum, James Hendler, and Geral Jay Sussman, 'Information accountability', *Commun. ACM*, **51**(6), 81–87, (June 2008).

[37] Peter Woolliams and David Gee, 'Accounting for user diversity in configuring online systems', *Online Review*, **16**(5), 303–311, (1992).