Anthropocentric Data Systems

Peter Triantafillou
University of Patras
peter@ceid.upatras.gr

1. THE CASE FOR A NEW VISION

Arguably, it all started with Mike Dertouzos' vision on the Information Marketplace [2]. Then, an explosion occurred. Social networks. Social computing. Social software. Groupware. Shareware. Open-source software. Personalized query answering and personalized information systems. Tagging. Folksonomies. Log and clickstream mining. Recommender systems. Crowdsourcing. Human-in-the loop and human-centered systems. Provably, these buzzwords have dominated the academic landscape within the data systems (and not only) community. There is a fundamental paradigm shift going on here. The old world, where the human was simply a passive user, has given way to a new world where humans contribute data, (storage, communication, and compute) resources, and software. Further, recently, humans take on tasks that actually alleviate and improve the jobs performed by machines and recent research from different domains have started looking into this realm where humans and computers share tasks, collaborating to achieve goals [4, 12, 11, 6, 1]!

Example endeavors from big players like that of RedHat Linux and Mozilla (for open-source software) and IBM (for groupware and shareware), have shown how humans can be much more than simple, passive users: they can play an active, key role in the design and implementation of the system itself. The peer-to-peer R&D and related products showed how the 'system's' resources can be made-up from users' actually collectively contributing their own resources. The next wave of web 2.0 and social networking and systems escalated user-contributed data collections and their sharing to unprecedented levels. Ditto for user networks, which became both data-resources to be discovered, and resources to be exploited for discovering other (meta)data of interest. Increasingly, users have been moving closer to the centre of the big picture! Further, lately companies like Innocentive (www.innocentive.com), Prosper (www.prosper.com), and Indifex (www.indifex.com) showed how crowdsourcing tasks requiring intelligence can lead to large value-added services.

The Vision (in brief). For the last years we have been following these lines aiming for a greater goal, described with the term *Anthropocentric Data Systems (*ADS*)*. At the highest level, *ADS* departs from the view that computers (at the centre) serve humans (at the periphery). Our vision embeds humans at the centre, exploiting human skills that can uniquely solve challenging problems, much better than any automata: *ADS* is about exploiting human intelligence and, in particular, collective human intelligence in order to help architect, design, and implement better systems.

First key observation: Humans can perform greatly in certain tasks individually and especially collectively (and several industries are actually banking on this). Think of tasks that require predictions; the issues related to crowdsourcing predictions is receiving a great deal of attention (e.g. [5, 7]) and significant steps forward have been made. Realizing that many of the internals of nowadays data systems depend on such tasks, why not facilitate collective human contributions that can yield much higher-performing internal algorithms and structures?

Let us think of fundamental components of current data systems. Cache management includes caching and prefetching algorithms. Deciding which cached items to replace and which items to prefetch is clearly a task that can greatly benefit from accurate data item popularity estimations. Query optimization algorithms typically depend on estimations of various statistical metrics, first on data items (e.g., on set cardinalities, selectivities of operators), on workload characteristics, IO device service times, etc. Again, accurate predictions of these can make a big difference in the system's performance. Indexing strategies (e.g. which items to index and how), can also be based on crowdsourcing predictions with respect to workload characteristics (identifying which items are popular and how they are accessed). The same holds for data placement algorithms on (distributed or centralized) storage devices: estimating the same key workload characteristics can be key to load balancing storage devices and thus ensuring high throughput. Concurrency control and consistency, is another area of application. As one example, optimistic concurrency control methods can be deployed if it is accurately predicted that (at least for some data items) conflicts are rare. Replication degrees and replica consistency again can greatly benefit from relevant predictions with respect to device/server lifetimes, conflict rates, etc. Clearly, the ability to make relevant predictions can be instrumental in improving the performance and functionality of several system components.

Second key observation: What we have learned from web 2.0 and social networking so far, points to novel re-

Proceedings of the VLDB Endowment, Vol. 4, No. 12

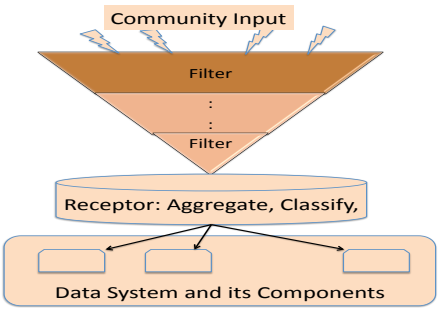

Figure 1: ADS Overview.

quirements for data modeling as a result of human activity. Obviously, the attributes and dimensions of interest of data items change dynamically (e.g., with users' tagging activities). But more fundamentally, *even what constitutes a data item itself can change also* as associations between data items, and/or between users, and/or between data items with users, can lead to newer higher-level data (information) items. And, interestingly, these items can themselves be queried for and/or form the bases of new access plans or indices used to derive the results of traditional queries.

What is *ADS* (in brief). At first, *ADS* is about research that develops a *conduit* that will import the input of humans into the internals of data systems. Such a conduit comes equipped with a number of *filters* and a *receptor*. Filters aim to remove "bad" feedback. For instance a filter may be applied to determine relevance and/or the expertise of the contributor. Another, may be applied to determine the trust the system places on the reliability of the contributor (e.g. to avoid maliciousness). Receptors aim to aggregate, classify, and direct such feedback to the appropriate system components, (figure 1). This includes classification based on affected system components, aggregations required for proper predictions, vote collections, etc.

Second, *ADS* is about research which discerns those tasks for which collective human input can actually improve system internals, a la task-dependent admission control functionality. Third, *ADS* is about researching the system architecture and structuring principles that will enable this fusion of automation and human feedback into the appropriate system internals. Finally, *ADS* is concerned with modeling human behavior and input, based on which such feedback can be anticipated, appropriately evaluated, and best exploited, as well as with appropriate interface systems, necessary to engage, visualize, and structure human input.

What *ADS* is not. *ADS* is not (just) a human-centered system [3] in that we do not just aim to make system functioning seamless and transparent to humans. *ADS* is also not (just) about *crowdsourcing* [4, 6, 1] in the sense that *ADS* is not just about offloading system tasks to humans and managing the system-human collaboration. (In other communities (e.g. the control automation community) crowdsourced systems are also known as *human-in-the-loop systems* [12, 11]). As mentioned above, *ADS* takes a broader view and a unique emphasis in having human inputs *become* the centre of the system. Humans thus actively, collectively, and dynamically define *(i)* the system internal functioning and *(ii)* the fundamental information units, their associations, and structuring.

2. CHALLENGES

Since *ADS* depend on human feedback it is crucial to incentivize it, aggregate it, filter out malicious/irrelevant inputs, identify which system components it affects, reconcile conflicting feedback, and provide system structuring and organization principles that facilitate these tasks and allow various system-component versions to coexist and be employed for different user communities/workloads.

2.1 Facilitate, Manage, Organize Feedback

A primary challenge pertains to the creation and selection of the best structure for the conduit, using which feedback is passed to system. Challenges here entail the architecture of the conduit itself, deciding on the structuring of the various filters and receptor and related algorithms.

Feedback. The form of feedback itself is an issue. How will this best facilitate user engagement? At the same time, a critical question is whether feedback should be explicitly requested by the system, or system monitoring of user behavior will be analyzed and appropriately evaluated and exploited. And, if explicitly requested, should the system engage all its users or a specific fraction of them, based on specific properties exhibited? The latter may have a critical impact on issues of reliability, security, and feedback independence raised below.

Filtering. A subset of the available filters must be provided in advance and decisions have to be made as to which they will be. Different filters may be needed for different systems layers. Also, a tree-like structuring of filters, based on the more-to-less general functionality may be used. Dynamic filter assignment, as input is provided, will decide on the appropriate filters to be applied to the input, before it reaches the receptor.

Trust. How the system places trust on the contributor is a big issue. Note that input should *not always* be limited to experts. There is preliminary evidence showing that for certain tasks, at least, such systems perform poorly when compared against systems where input was accepted from independent non-experts. Yet, for other tasks, it is evident that the contributor's expertise should be weighted favorably [5]. Hence, trust-expertise filters should be applied with caution. This duality must be dealt with.

Maliciousness. Avoiding malicious user feedback is a critical issue. First, note that many prediction mechanisms incorporate mechanisms for incentivizing truthfulness [7, 5]. Further, there exist already significant contributions in the direction of identifying trustworthy contributors from the networking and P2P communities. Interestingly, promising advances in this realm involve information about users that is readily available/extractable from their social networks [14]. This is a good example of how monitoring user behavior (e.g. their social links in a social network) can provide implicit feedback for the filters themselves. But clearly much more is required.

2.2 Aggregate, Classify, and Route Feedback

Aggregation of feedback has two dimensions. The first concerns the ability to form a complete view, or as large as possible view of "the true state of the world". For our purposes this implies feedback on a large percentage of the possible affects some policy may have. This implies the need for *independent* sources of feedback. The second dimension concerns how to best aggregate individual input. Here, one

must first deal with conflicting feedback and resolve such conflicts. Should a voting method be used for this? If so, which voting aggregation method is most appropriate?

A sensitive issue when aggregating feedback emerges when considering the goal of personalized information access. For this to be achieved, individual input should be aggregated perhaps multiple times for multiple reasons: the danger exists that individual input and its value for facilitating personalized accesses will be lost if it were to be simply aggregated into a single "big" aggregate. This leads to a "multi-modal" aggregation method, dealing with the dualism of personalized and community aggregates.

Classification of feedback is intended primarily to decide to which components specific input pertains. User input can affect system components both directly and indirectly. Central questions here concern how to structure and model the dependencies of system components. The definition of the *dimensions of interest*, for which input is sought, will help. Specific feedback spans one or more dimensions of interest. Each dimension of interest concerns one or more system components, at various degrees. Complementarily, feedback classification aims to identify sought feedback and its potential for overall impact, based on the model of system-component-user-input dependencies.

New feedback must be compared across previous input. This aims to identify congruent and conflicting feedback. It is possible that classification may best be applied on feedback aggregations. And, aggregators may best be applied on classified input.

Probabilistic data management may be instrumental in managing the synergy between the filters and the receptor. Clearly, different confidence will be placed on different human inputs. Abstractly, a probabilistic relational model associating filtered and aggregated input with the prediction random variables (attributes) and related confidence values is of value for ADS. Determining which input will affect which system components can be thought of applying the proper aggregation queries over such probabilistic tables.

Feedback routing is the process that determines the flow of feedback towards the affected components. Think of networks, where nodes represent system components, edges represent input-dependencies, and the routing algorithm determines how to best route the continuous stream of input through the network so to achieve best system performance.

2.3 Architectural, Structuring Principles

So how do we go about facilitating the design and implementation of *ADS* systems? First, note that *ADS* are "multiple-personality" systems. Consider different "conflicting" user communities, with different usage scenarios and classifying, mutually-congruent feedback, into several clusters representing "conflicting" communities. In essence, this translates to having multiple versions of system components, one for each optimization goal (user community).

We envision a **Faceted System Architecture**. *Facets* embody different component versions.

Each facet can be defined as a clustered sets of services, implementing a version of the *ADS* system, according to an aggregated set of congruent votes/feedback. The end goal being that incoming requests will be properly associated with specific facets and the correct functionality will be offered. Complementarily, if facets are "competing", only one facet, representing the dominant feedback is active while the others remain dormant, until the aggregated feedback suggests that they become active. Therefore, facets need be defined in association with the aggregation and classification activities explained previously. A key issue is how to collect and aggregate/cluster congruent feedback, on which a facet will be based? i.e., we need a mapping function of a feedback cluster onto a facet. An interesting proposal is to define and install facets dynamically, in response to emerging relevant feedback. For this, one can envision a set of predefined such facets, which remain dormant until the appropriate feedback emerges. Going a step further, new facets may be crowdsourced dynamically, with the requirements emerging from the aggregated user feedback.

Query optimization. How are facets engaged during query execution? How do we decide on which facet to employ? Consider, for example different facets existing for different personalizations/scenarios. Suppose, there is no available evidence (from the current query) on which facet to employ. How does the *ADS* system process this query? Can the users profile be somehow matched against facet profiles, using some similarity distance measure? Should there be a default facet executed for run-of-the-mill queries and specialized facets engaged for higher-priority users/queries (e.g., from paying customers)? How can we model/predict the cost of offering a functionality associated with a given facet and incorporate this cost into the query optimization objectives? Related to these query optimization issues, are the issues concerning the cost estimation for maintaining and/or creating facets including their crowdsourcing).

2.4 Predictions, Judgments on System Tasks

Crowdsourcing predictions is already a well-studied field with many applications in economics (eg predicting expected benefits of a policy), meteorology (weather predictions), medicine (predicting treatment outcomes), etc. Within *ADS*, predictions and judgements refer to the fundamentals of system-component operation. The question here is to identify the key items calling for prediction crowdsourcing, for specific system components, such as query optimization, caching, data placement, fault tolerance, etc. Are specialized prediction models (with algorithms, and incentives) needed for these tasks? Can a single prediction model suffice for all system components?

Incentives play a crucial role here. However, the *ADS* environment is different. The pay-off process for being "truthful" is more complicated (e.g., not a as simple as in a betting game, where predicting correctly can accrue more money into the better's pocket). Referring to (heavy) users of the system, pay-offs can take the form of more resources allocated and/or higher priorities being associated when users' jobs run. (Interestingly, this implies an association of the prediction mechanisms with system components responsible for resource allocation - this is by itself an interesting research problem). But, for algorithms/system experts, incentivizing truthfulness is a completely open problem.

Relevant Tasks Identification. For what tasks should input be sought after? We have referred to "prediction" tasks earlier, and some of the fundamental system components which could benefit from crowdsourced, system-incorporated predictions. Even for such prediction tasks, what are the 'random variables' for which we wish to apply our prediction strategy, in each component's case? Which system tasks are amenable to fundamental prediction re-

quirements, such as the *independence* of contributors with private evidence, *common priors* (that is, subjective understandings among contributors of the basics of the to-be-predicted variable), and rational contributor's behavior? Essentially, we must research which are the system tasks and how they depend on estimations, predictions, and judgements for which the so-called "crowd intelligence", "crowd wisdom", or "community wisdom" can be most appropriate.

2.5 User Modeling and Interfaces

Human behavior models are needed for modeling input rates and types, as well as service times and rates for tasks assigned to humans. For the latter, there has been some progress, modeling service times for human operators, using queueing networks of tasks with human operators providing the service [11]. And such models need be coupled with interfaces that will help elicit and guide human feedback.

3. ADS INSTANCES

The *ADS* instances that follow mainly refer to the second key observation made in Section 1.

We define **Organic Data Systems** as systems, which (i) develop naturally, (ii) whose contents are not viewed using a predetermined, static manner, and (iii) whose users view the entire set of entities of interest for their information need, rather than just isolated data items. This is in dark contrast to either DB schemas and hierarchical file system views and query results based on such schemas and files. The focus of this work is on defining what are the "data items", based on users contributions and descriptions of them. Essentially, it is the user community that defines the data space(s), which the system will be called to organize and access.

Starting with primitive data items, the user community, and the dynamic user descriptions of data and users, relations between these entities are discovered. This leads to the formation of new complex and query-able information units. The data system provides an ever-growing data space of sets of information units and relationships between them. Primitive data items can be discovered by users (after issuing queries). Users characterize these, add relationships to other units, and define new complex information units, (consisting of these units and their relationships), which can be themselves part of the answer to future queries, etc. Hence, users define which are the data items of interest and with their help the system defines indices for the efficient access to this multi-granularity data space [13].

Decentralized Social Networks. With **eXO** [10] we provided a decentralized social networking system, primarily motivated by the desire for decentralization (i.e., eliminating the need for powerful players who store, exploit, control, data items, user profiles, and user associations, without much regard for the individual human contributors). What makes *eXO* an *ADS* system is that it allows data owners to control where their information is stored/replicated, and how it is described and indexed in the system. In this, owners take into account the community wisdom as to how best tag and index own items and who can access them and how.

Crowdsourced Taxonomies and Ranked Retrieval. In this effort [8] human input guides the construction of thematic taxonomies. Specifically, users perform extended tagging, explicitly providing IS-A relationships between their tags. The system aggregates these into a taxonomy, resolving conflicts and exploiting community wisdom. In turn, these taxonomies are employed as the *sole* index for all documents. Novel algorithms are then developed for ranked retrieval of data items described with crowdsourced taxonomy nodes. This setup can do away with the need for constructing, storing, maintaining, and utilizing expensive inverted text indices for documents [9].

4. CONCLUSIONS

The time is ripe. Never before have we witnessed such a huge-scale of human involvement in Information Systems. Complementarily, various needed R&D results are rapidly emerging from several scientific communities (from DBs and IR, to Machine Learning and Data Mining, to Distributed Systems and Networks, to Software Engineering, to Control Theory, to Economics, Psychology and Sociology, etc). The paths before us present interesting routes, opening up the scene for many profound research accomplishments. Hopefully, the biggest one will be reaching the end of the road, where collective human wisdom will be embodied and thus become instrumental in developing the next paradigm for systems creating, retrieving, and managing data.

5. REFERENCES

[1] O. Alonso and M. Lease. Crowdsourcing 101: Putting the "wisdom of the crowd" to work for you. *WSDM Tutorial*, 2011.

[2] M. Dertouzos. *What will be*. HarperEdge, 1998.

[3] M. Dertouzos. *The Unfinished Revolution: Human-Centered Computers and What They Can Do For Us*. HarperBusiness, 2001.

[4] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: Answering queries with crowdsourcing. *In: Proc. ACM SIGMOD Conference*, pages 517–528, 2011.

[5] S. Goel, D. M. Reeves, and D. M. Pennock. Collective revelation: A mechanism for self-verified, weighted, and truthful predictions. *In: Proc. 10th ACM Conf. on Electronic Commerce*, pages 265–274, 2009.

[6] P. Ipeirotis. Managing crowdsourced human computaiton. *WWW Tutorial*, 2011.

[7] R. Jurca and B. Faltings. Incentives for expressing opinions in online polls. *In: Proc. 9th ACM Conf. on Electronic Commerce*, pages 119–128, 2008.

[8] D. Karampinas and P. Triantafillou. Crowdcourcing taxonomies. *in preparation*, 2011.

[9] I. Kontotasiou and P. Triantafillou. Treats: Optimal ranked retrieval on tag taxonomies. *submitted.*, 2011.

[10] A. Loupasakis, N. Ntarmos, and P. Triantafillou. exo: Decentralized autonomous scalable social networking. *In: Proc. 5th CIDR*, pages 85–95, 2011.

[11] K. Savla and E. Frazzoli. A dynamical queue approach to intelligent task management for human operators. *In: Proc. of the IEEE (to appear)*, 2011.

[12] K. Savla, T. Temple, and E. Frazzoli. Human-in-the-loop vehicle routing policies for dynamic environments. *In: Proc. IEEE Conference on Decision and Control*, pages 1145–1150, 2008.

[13] P. Triantafillou, A. Pasiopoulos, and N. Ntarmos. Organic file systems. *in submission*, 2011.

[14] B. Viswanath, A. Post, K. Gummadi, and A. Mislove. An analysis of social network-based sybil defenses. *In: Proc. ACM SIGCOMM*, pages 363–374, 2010.