user preferences [16]. Other methods showed how people can also help to improve the quality of search engines [21] and complete missing information in social networks, such as tags associated with its members [4]. We have focused in this paper on the data layer of a particular kind of crowdsourcing games, namely *datasourcing* games that aim to use the collective wisdom to construct a large database of facts.

Much research has been recently directed in the databases community to the development of DB platforms that allow for declarative specification of the crowdsourced data components [13, 23]. These platforms are providing declarative lagunage support and tools to define what data will be retrieved from the crowd (the choice of questions to ask players in our case), for example by adding a *CROWD* operator to a certain column in a *Create* SQL statement of a table [13] and how the flow of such tasks should be managed [23]. While the choice of data to ask the crowd for can be done declaratively in these frameworks, the other tasks described in this paper such as aggregating the data (and deciding which data is correct), choosing which users to ask, assigning scores to users etc. are not addressed or done in a hard-coded manner (see, for example, the *Combiner* operator implemented as *MajorityVote* in [22]). Our platform allows declrative formulation of policies for these tasks and is therefore complementary to the platforms presented above.

Policies for determining correctness of data in presence of contradictions often appear in the context of *data cleaning*. We have already mentioned some data cleaning policies and showed how to implement them using our framework. Many other policies have been proposed in the literature: In [3] the authors present a different approach for cleaning by using string-transformation rules for correcting errors in the data. [5] presents a technique to solve key violations using probabilistic choice over possible Database repairs. [24] discusses techniques for evaluating the trustworthiness of information sources. These solutions are all hard-coded, in contrast to the generic declarative framework that we propose here.

Information integration often entails fusion of data from various sources (e.g, [6]). This requires the identification of common objects and the resolution of possible conflicts. Such (possibly probabilistic) data fusion algorithms may also benefit from the declarative framework described in the present paper and we intend to study its application to this domain in future work.

Last, we note that there are some declarative frameworks that support queries on probabilistic data (e.g. [1, 17, 12, 18]). In particular, in [18] the authors present a declarative framework for probabilistic rules, based on Markov Chains. However, all of these works do not allow the definition of recursive rules, hence for example cannot express the PageRank-style cleaning rules described here.

7. CONCLUSION

We presented a novel declarative framework for the data management layer of data-sourcing games. At the core of our framework is a declarative language that allows to express probabilistic and recursive policies, which we demonstrated to be useful for different data management aspects of such games. We further described implementation issues addressed when putting the platform into practical use, in the context of the Trivia Masster game, and reported the results of our experimental study with respect to the system.

The design of dedicated optimization algorithms for the framework is a challenging future research. In particular, we intend to study incremental maintenance for rapidly adjusting to changes and additions to the facts database, thereby avoiding a full off-line (re)evaluation.

8. REFERENCES

[1] P. Agrawal, O. Benjelloun, A. Das Sarma, C. Hayworth, S. U. Nabar, T. Sugihara, and J. Widom. Trio: A system for data, uncertainty, and lineage. In *VLDB '06*.

[2] L. Antova, C. Koch, and D. Olteanu. "Query Language Support for Incomplete Information in the MayBMS System". In *Proc. VLDB*, 2007.

[3] A. Arasu, S. Chaudhuri, and R. Kaushik. Learning string transformations from examples. *PVLDB*, 2(1), 2009.

[4] M. Bernstein, D. S. Tan, G. Smith, M. Czerwinski, and E. Horvitz. Collabio: a game for annotating people within social networks. In *UIST '09*. ACM.

[5] G. Beskales, I. F. Ilyas, and L. Golab. Sampling the repairs of functional dependency violations under hard constraints. In *VLDB '10*.

[6] J. Bleiholder and F. Naumann. Declarative data fusion - syntax, semantics, and implementation. In *ADBIS*, '05.

[7] D. C. Brabham. Crowdsourcing as a Model for Problem Solving: An Introduction and Cases. *Convergence*, 14(1), 2008.

[8] M. K. Cowles and B. P. Carlin. Markov chain monte carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91, 1996.

[9] N. Dalvi and D. Suciu. "Efficient query evaluation on probabilistic databases". In *Proc. VLDB*, 2004.

[10] D. Deutch, O. Greenshpan, B. Kostenko, and T. Milo. Using markov chain monte carlo to play trivia (demo). In *ICDE*, 2011.

[11] D. Deutch, C. Koch, and T. Milo. On probabilistic fixpoint and markov chain query languages. In *PODS '10*.

[12] R. Fink, D. Olteanu, and S. Rath. Providing support for full relational algebra in probabilistic databases. In *ICDE*, 2011.

[13] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: answering queries with crowdsourcing. In *SIGMOD Conference*, 2011.

[14] D. Freedman. *Markov Chains*. Springer-Verlag, 1983.

[15] A. Galland, S. Abiteboul, A. Marian, and P. Senellart. Corroborating information from disagreeing views. In *WSDM '10*.

[16] S. Hacker and L. von Ahn. Matchin: eliciting user preferences with an online game. In *CHI '09*.

[17] J. Huang, L. Antova, C. Koch, and D. Olteanu. Maybms: a probabilistic database management system. In *SIGMOD Conference*, 2009.

[18] R. Jampani, F. Xu, M. Wu, L. L. Perez, C. Jermaine, and P. J. Haas. Mcdb: a monte carlo approach to managing uncertain data. In *SIGMOD '08*.

[19] C. Koch. Approximating predicates and expressive queries on probabilistic databases. In *PODS '08*.

[20] N. Leone and et al. "The INFOMIX system for advanced integration of incomplete and inconsistent data". In *Proc. SIGMOD*, 2005.

[21] H. Ma, R. Chandrasekar, C. Quirk, and A. Gupta. Improving search engines using human computation games. In *CIKM '09*.

[22] A. Marcus, E. Wu, D. R. Karger, S. Madden, and R. C. Miller. Human-powered sorts and joins. *PVLDB*, 5(1), 2011.

[23] A. Marcus, E. Wu, S. Madden, and R. C. Miller. Crowdsourced databases: Query processing with people. In *CIDR*, 2011.

[24] A. Marian and M. Wu. Corroborating information from web sources. *IEEE Data Eng. Bull.*, 34(3), 2011.

[25] Amazon's mechanical turk. https://www.mturk.com/.

[26] C. Re, N. Dalvi, and D. Suciu. Efficient top-k query evaluation on probabilistic data. In *ICDE '07*.

[27] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag, Inc., 2005.

[28] Shannon and Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, 1949.

[29] Q. Su, D. Pavlov, J.-H. Chow, and W. C. Baker. Internet-scale collection of human-reviewed data. In *WWW '07*.

[30] Top coder. http://www.topcoder.com/.

[31] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *CHI '04*.

[32] L. von Ahn and L. Dabbish. Designing games with a purpose. *Commun. ACM*, 51(8), 2008.