Towards Socio-Technical Design of Explicative Systems: Transparent, Interpretable and Explainable Analytics and its Perspectives in Social Interaction Contexts

Martin Atzmueller

Tilburg University, Department of Cognitive Science and Artificial Intelligence,
Warandelaan 2, 5037 AB Tilburg, The Netherlands
m.atzmuller@uvt.nl

Abstract. This paper outlines an approach towards socio-technical design of explicative systems enabling transparent, interpretable and explainable analytics. We sketch the TIE approach for the socio-technical explicative system design and discuss its fundamental principles and perspectives. Furthermore, we exemplify the application of the proposed approach in social interaction contexts.

Keywords: Explainable AI Systems, Affective Computing, Social Interactions

1 Introduction

The analysis of social interactions, e. g., in AI systems and affective computing contexts, has emerged as a prominent research direction in order to investigate, for example, their structure, dynamics and predictability. Social (human) interactions, e. g., including group interactions as well as connections mediated via sensors can then be modeled in social interaction networks, e. g., [1–3], also towards intelligent systems providing analytics capabilities in complex domains, e. g., [4–7]. Examples of such networked system contexts include interactions through social media, co-authorships between researchers and connections of smart devices [7]. Moreover, with the emergence of Affective Computing, the Internet of Things and the involved ubiquitous devices, we have access to multi-modal social interaction datasets which can be modeled as networks.

However, typically standard methods for analysis neglect important aspects relating to transparency, interpretability and explainability – which is necessary for comprehensive human computing – targeting and enabling the human-in-the-loop. In particular, appropriate approaches and methods require transparency on models and results, their interpretability, and appropriate explanation-aware techniques e.g. for increasing the acceptance of the patterns and their evaluation [8–10].

This paper outlines first perspectives on designing computational methods taking into account cognitive, human–machine and computational requirements in an integrated way: We outline and discuss imporant concepts of transparent, interpretable and explainable analytics of social interactions, e. g., referring to [8, 9, 11–15]. Furthermore, we sketch an approach for the socio-technical design of explicative systems enabling transparent, interpretable and explainable analytics of social interactions.

2 Related Work

In the following, we briefly discuss related work concerning computational systems enabling transparent, interpretable and explainable analytics, before we turn to related socio-technical design approaches.

2.1 Transparent, Interpretable and Explainable Analytics

Recently, the concept of transparent and explainable models has gained a strong focus and momentum in the machine learning and data mining community, e. g., [16–18]. In particular in the scope of black box methods, further explanation and interpretation are required to enable domain experts and users to understand, trust, and transform the novel and useful model into a real-world application [19, 20]. This is in particular relevant in human-machine systems, which are prominent in the fields of affective computing, human computing, and social computing. Explicative analytics, as introduced by [9], is a comprehensive paradigm for interpretable, transparent and explainable data analytics, i. e., machine learning and data mining. It aims to describe and explain the underlying structure of the data, by using explanation-aware methods. Similar to the philosopical process of *explication* cf. [21, 22] which aims to make the implicit explicit, explicative analytics aims to model, describe and explain the underlying structures.

Exemplary approaches focus on specific models, e. g., tree-based [23] or pattern-based approaches [24]. Here, also, methods for associative classification, e. g., class association rules [25] can be applied for obtaining explicative, i. e., transparent, interpretable, and explainable classification models [9]. Then, individual steps of a classification (i. e., a decision can be traced-back to the model, similar to *reconstructive explanations*, cf. [26] on several explanation dimensions [16]. For model agnostic explanation, e. g., [27], general directions are given by methods considering counterfactual explanation, e. g., [28, 29], data perturbation and randomization techniques as well as interaction analysis methods, e. g., [30].

For the analysis of social interactions, we need to consider further perspectives concerning the multi-relational nature and complexity of the respective data and information: Traditional approaches for community detection, for example, aim at partitioning the social network graph. The richer available data in complex networks, i.e. additional information of each user, e. g., regarding demographics, interests, help us develop mining methods which can take advantage them and detect interesting communities, associated with interesting and interpretable descriptions in terms of user information [31]. In case of the link prediction, the prediction of the topological evolution of a network over time is concerned, for which topological as well as user information can be utilized.

Furthermore, previous work focused on finding a "perfect" set of features capable of predicting the formation of a link, most of the times with a "black box" type of approach. However, grouping the features on their topological scope [32] leads to a efficient and explainable set of features, capturing the essential network properties. Another exemplary major social network analysis application is the development of recommendation systems [33, 34]. Explainable recommendation methods that not only predict a numerical rating for a recommended item, but also generate explanations for the users' preferences, improve effectiveness, transparency, and user trust [33–35].

2.2 Socio-Technical Design Methods

Transparency and explainability are crucial for human-machine computational systems, e. g., in the fields of affective computing, human computing, social computing etc. There are different complementing perspectives providing important aspects. For example, the respectively applied intelligent and pervasive systems that are being used at a human and societal level need to be transparent and to be able to explain their decisions to the respective stakeholders and actors. In addition, by exploiting such explicative capabilities, then the provided interpretability and explainability will also enhance trust in the system at the level of the users, and due to that improve the quality and performance of human-machine systems. In the design and development of such systems, therefore both the computational as well as the explicative requirements need to be taken into account. Then, functional as well as normative requirements come into play.

One prominent method for matching normative (e. g., legal) requirements with technical requirements and specific implementation options, is the KORA method [36, 37]. It has been applied in several contexts, e. g., in an integrated approach for socio-technical design and development of ubiquitous computing applications [6,38,39], e. g., in the context of designing a ubiquitous and social conference guidance system [40]. The basic KORA method aims at acquiring technical implementations based on legal requirements. It is built on a four step process model, starting with legal requirements that are mapped to legal criteria which are then matched with functional requirements. The legal requirements are typically derived from application specific legal provisions, e. g., given by the European Union's new General Data Protection Regulation (GDPR), which also enforces a "right to explanation" (regarding specific algorithmic decisions) [41]. These legal provisions are then made more concrete in the second step, also including technical functions as well as legal and social aspects, formalized in specific criteria. These criteria are then mapped to functional requirements in the third step, e. g., supported by domain experts or based on utilizing design patterns. In the final fourth step, these functional requirements then map to specific implementation choices. Relating to common process models in software engineering, the first three steps basically aim at requirements analysis, whereas the last step involves specific methods, patterns, and techniques relating to the concrete instantiations and implementations.

A design method integrating several socio-technical components besides only legal requirements is the *VENUS approach* [6, 42–45]: It is a design methodology that supports the development of socially acceptable UC applications. Here, applications should not only satisfy a given set of functional requirements but also comply to the given set of user requirements – relating to concepts in human computing – i. e., those given in terms of usability [46] and trust [47–49], also including legal regulations [50]. For a detailed discussion of the VENUS approach we refer to [6, 42, 44]. In contrast to more general methods like, for example, design thinking [51, 52] or agile approaches like SCRUM [53, 54], the VENUS approach targets more specific applications, i. e., ubiquitous computing. While agile methodologies can be implemented in the different steps and phases, the major difference to those more general approaches is to focus on integrating/combining/merging normative and functional requirements into account utilizing core concepts of the KORA method. This is also the approach we are proposing towards the design approach for explicative systems sketched in the next section.

3 Towards Designing Explicative Analytics Systems

Most common data analysis methods on social interactions lack important aspects, i. e., *interpretability*, *transparency* and *explainability* in order to be *explicative* towards its users. Especially considering complicated black-box models this becomes relevant, e. g., when providing recommendations. Here, intransparent methods and models make it more difficult to spot mistakes and can lead to biased decisions; in general, they stretch the trust humans have (and should rightfully have) in the respective predictions. This is in particular relevant for the domain of affective, human and social computing, where the interactions between human and machine are fundamental. Below, we first introduce the general approach, before we exemplify its application in social interaction contexts, specifically targeting explicative link prediction.

3.1 TIE Approach for Designing Explicative Analytics Systems

Explicative analytics targets interpretable (and transparent) models utilizing exploratory and explanation-aware methods. These can be constructed and inspected on different layers and levels. Then, such methods can be mapped to technical specifications according to the given criteria – corresponding to components of transparency, interpretability and explainability, e. g., using the core KORA method. We adapt this idea in the *TIE* approach. – for the development of computational sytems enabling transparent, interpretable and explainable analysis of social interactions. The TIE approach is based on the *VENUS* approach – specializing it towards components in the scope of transparency, interpretability and explainability. At its core, the TIE approach features an iterative development process: It consists of *context analysis*, (conceptual) *requirements analysis*, *method design* and *implementation*, and finally *method evaluation*. The process starts with a method application scenario – in the scope of analyzing social interactions. This scenario is used to elicit requirements in the analysis – for designing the respective analytics method(s). These requirements are derived from concepts in transparency, interpretability, and explainability of the involved analytics method. In order to construct a comprehensive method integrating those, they are merged into a concept design of the respective method. In the next phase, the method is designed and implemented. Finally, the developed method is evaluated, yielding results, and specifically recommendations for improvement. Based on the obtained results, a new iteration cycle of the *TIE* development process can then be initiated taking the results/recommendations into account for refining the design concepts.

Considering that, it is important to note that, as is known from requirements engineering, if we consider the antecedents of a given latent construct - such as the dimensions of transparency - then, this can be interpreted as a under-specified functional requirement [33]. Thus, when these requirements are considered during system design, they need to be translated into functional requirements. As outlined above, the TIE approach features a process model for turning such requirements into the implementation. Then, for example, normative requirements from law or ethics can be handled in an integrated manner. Below, we describe some examples of that transformation and implementation.

3.2 Explicative Link Prediction

As a prominent example in the context of analyzing of social interactions, we can consider the method of link prediction, to be implemented in an explicative system. Then, as a general requirement, every decision of the system (regarding link prediction) should be traceable. As a concrete instantiation, this requires that link predictions should be explained (i. e., justified). Then, in the concept design of the according method, we can then rely on established principles in explanation-aware computing. For creating explanations, for example, we refer to the model introduced by Roth-Berghofer and Richter [15]. In a general explanation scenario we have three main participants [15]: the *user* who is corresponding with the software system via its user interface, the *originator*, i.e., the tool that provides the functionality with which the user wants to solve a task, and the *explainer*. For constructing the explanations, the explainer can then utilize the structures implemented in the model used for representing the social interactions in the targeted domain. In our case, we utilize the available network and link structures in order to enable reconstructive explanations [26]. As a concrete implementation of that method, we can resort to knowledge-based techniques enabling traceable link prediction. Using Answer Set Programming (ASP) [55], for example, such prediction can be implemented [56], and according explanations can then be generated [57] accordingly.

4 Conclusions

Transparency, interpretability and explainability are key features for developing web systems that comply to requirements from, e. g., cognitive, human–machine and human (computational) perspectives. Currently, the work in these areas is only yet starting, however in the light of large-scale (social) effects of the developed methods, appropriate methods supporting to take the above mentioned requirements into account are crucial.

This paper sketched an approach for the development of explicative systems enabling transparent, interpretable and explainable analytics of social interactions. We outlined the process and its components. In a social system with several methods for analyzing social interactions, for example, explicative analytics methods are crucial in order to incorporate user requirements, i. e., enabling transparency, interpretability and explainability – e. g., for fostering trust in the system.

Refining our initial perspectives, we aim to apply the proposed approach in the context of a set of methods and systems in the context of artificial intelligence and affective computing. Furthermore, for future work, we aim to further refine the TIE approach, e. g., by abstracting the given requirements into specific design patterns, and thus working towards providing refined recommendations during the development process. Important classes of patterns, for example, consider the complexity of the explanations, customized explanation and transparency levels as well as specific ethical requirements, e. g., [35, 58, 59]. Furthermore, we aim to apply the process in different domains, e. g., also towards business and industrial systems capturing a variety of interactions to be analyzed in complex artificial intelligence systems.

References

1. Mitzlaff, F., Atzmueller, M., Benz, D., Hotho, A., Stumme, G.: Community Assessment using Evidence Networks. In: Analysis of Social Media and Ubiquitous Data. Volume 6904 of LNAI. (2011)
2. Mitzlaff, F., Atzmueller, M., Stumme, G., Hotho, A.: Semantics of User Interaction in Social Media. In Ghoshal, G., Poncela-Casasnovas, J., Tolksdorf, R., eds.: Complex Networks IV. Volume 476 of Studies in Computational Intelligence. Springer, Berlin, Germany (2013)
3. Atzmueller, M.: Data Mining on Social Interaction Networks. Journal of Data Mining and Digital Humanities **1** (June 2014)
4. Puppe, F., Atzmueller, M., Buscher, G., Huettig, M., Lührs, H., Buscher, H.P.: Application and Evaluation of a Medical Knowledge-System in Sonography (SonoConsult). In: Proc. 18th European Conference on Artificial Intelligence (ECAI 2008). (2008) 683–687
5. Atzmueller, M., Becker, M., Doerfel, S., Kibanov, M., Hotho, A., Macek, B.E., Mitzlaff, F., Mueller, J., Scholz, C., Stumme, G.: Ubicon: Observing Social and Physical Activities. In: Proc. 4th IEEE Intl. Conf. on Cyber, Physical and Social Computing (CPSCom 2012), Washington, DC, USA, IEEE Computer Society (2012) 317–324
6. Atzmueller, M., Behrenbruch, K., Hoffmann, A., Kibanov, M., Macek, B.E., Scholz, C., Skistims, H., Söllner, M., Stumme, G.: Connect-U: A System for Enhancing Social Networking. In: Socio-technical Design of Ubiquitous Computing Systems. (2014)
7. Du, Z., Hu, L., Fu, X., Liu, Y.: Scalable and Explainable Friend Recommendation in Campus Social Network System. In: Frontier and Future Development of Information Technology in Medicine and Education, Springer (2014) 457–466
8. Atzmueller, M., Roth-Berghofer, T.: The Mining and Analysis Continuum of Explaining Uncovered. In Bramer, M., Petridis, M., Hopgood, A., eds.: Research and Development in Intelligent Systems XXVII, Springer London (2011) 273–278
9. Atzmueller, M.: Onto Explicative Data Mining: Exploratory, Interpretable and Explainable Analysis. In: Proc. Dutch-Belgian Database Day, TU Eindhoven, Netherlands (2017)
10. Atzmueller, M.: Declarative Aspects in Explicative Data Mining for Computational Sensemaking. In: Proc. International Conference on Declarative Programming, Springer (2018)
11. Schank, R.C.: Explanation: A first pass. In Kolodner, J.L., Riesbeck, C.K., eds.: Experience, Memory, and Reasoning, Hillsdale, NJ, Lawrence Erlbaum Associates (1986) 139–165
12. Gregor, S., Benbasat, I.: Explanations From Intelligent Systems: Theoretical Foundations and Implications for Practice. MIS Quarterly **23**(4) (1999) 497–530
13. Roth-Berghofer, T., Schulz, S., Leake, D., Bahls, D.: Explanation-Aware Computing. AI Magazine **28**(4) (2007)
14. Sormo, F., Cassens, J., Aamodt, A.: Explanation in Case-Based Reasoning–Perspectives and Goals. Artif Intell Rev **24**(2) (October 2005) 109–143
15. Roth-Berghofer, T.R., Richter, M.M.: On explanation. Künstl. Intelligenz **22**(2) (May 2008) 5–7
16. Atzmueller, M., Roth-Berghofer, T.: The Mining and Analysis Continuum of Explaining Uncovered. In: Research and Development in Intelligent Systems XXVII, Springer (2011) 273–278
17. Biran, O., Cotton, C.: Explanation and Justification in Machine Learning: A Survey. In: IJCAI-17 Workshop on Explainable AI. (2017)
18. Guidotti, R., Monreale, A., Turini, F., Pedreschi, D., Giannotti, F.: A Survey of Methods for Explaining Black Box Models. arXiv preprint arXiv:1802.01933 (2018)
19. Samek, W., Wiegand, T., Müller, K.R.: Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models. arXiv preprint arXiv:1708.08296 (2017)

20. Liu, S., Wang, X., Liu, M., Zhu, J.: Towards Better Analysis of Machine Learning Models: A Visual Analytics Perspective. Visual Informatics **1**(1) (2017) 48–56
21. Carnap, R.: Logical foundations of probability. (1962)
22. Maher, P.: Explication Defended. Studia Logica **86**(2) (2007) 331–341
23. Tolomei, G., Silvestri, F., Haines, A., Lalmas, M.: Interpretable Predictions of Tree-based Ensembles via Actionable Feature Tweaking. In: Proc. KDD, ACM (2017)
24. Duivesteijn, W., Thaele, J.: Understanding Where Your Classifier Does (Not) Work – The SCaPE Model Class for EMM. In: Proc. ICDM, IEEE (2014) 809–814
25. Atzmueller, M., Hayat, N., Trojahn, M., Kroll, D.: Explicative Human Activity Recognition using Adaptive Association Rule-Based Classification. In: Proc. IEEE Future IoT, IEEE (2018)
26. Wick, M.R., Thompson, W.B.: Reconstructive Expert System Explanation. Artificial Intelligence **54**(1-2) (1992) 33–70
27. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: High-Precision Model-Agnostic Explanations, AAAI (2018)
28. Mandel, D.R.: Counterfactual and Causal Explanation: From Early Theoretical Views To New Frontiers. In: The Psychology of Counterfactual Thinking. Routledge (2007) 23–39
29. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. (2017)
30. Henelius, A., Puolamäki, K., Ukkonen, A.: Interpreting Classifiers through Attribute Interactions in Datasets. Proc. ICML Workshop on Human Interpretability in Machine Learning (2017)
31. Pool, S., Bonchi, F., Leeuwen, M.v.: Description-Driven Community Detection. ACM Transactions on Intelligent Systems and Technology (TIST) **5**(2) (2014) 28
32. van Engelen, J.E., Boekhout, H.D., Takes, F.W.: Explainable and efficient link prediction in real-world network data. In: Proc. IDA, Springer (2016) 295–307
33. Hoffmann, A., Söllner, M., Hoffmann, H.: Twenty software requirement patterns to specify recommender systems that users will trust. In: 20th European Conference on Information Systems (ECIS), Barcelona, Spanien (2012)
34. Ren, Z., Liang, S., Li, P., Wang, S., de Rijke, M.: Social Collaborative Viewpoint Regression with Explainable Recommendations. In: Proceedings of the tenth ACM international conference on web search and data mining, ACM (2017) 485–494
35. Nalepa, G.J., van Otterlo, M., Bobek, S., Atzmueller, M.: From Context Mediation to Declarative Values and Explainability. In: Proc. IJCAI/ECAI Workshop on Explainable Artificial Intelligence (XAI-18), Stockholm, Sweden, IJCAI (2018)
36. Hammer, V., Pordesch, U., Roßnagel, A.: Betriebliche Telefon- und ISDN-Anlagen rechtsgemäß gestaltet. Edition SEL-Stiftung. Springer (1993)
37. Roßnagel, A., Hammer, V.: KORA. Eine Methode zur Konkretisierung rechtlicher Anforderungen zu technischen Gestaltungsvorschlägen für Informations- und Kommunikationssysteme. Infotech **1** (1993) 21 ff.
38. Behrenbruch, K., Atzmueller, M., Kniewel, R., Hoberg, S., Stumme, G., Schmidt, L.: Gestaltung technisch-sozialer vernetzung in der arbeitsorganisation: Untersuchung zur nutzerakzeptanz von rfid-technologie. In: GfA-Frühjahrskongress, Chemnitz (2011)
39. Geihs, K., Leimeister, J., Roßnagel, A., Schmidt, L.: On socio-technical enablers for ubiquitous computing applications. In: Proc. Workshop on Enablers for Ubiquitous Computing and Smart Services, Izmir, Turkey, IEEE (Juli 2012) 405–408
40. Atzmueller, M., Benz, D., Doerfel, S., Hotho, A., Jäschke, R., Macek, B.E., Mitzlaff, F., Scholz, C., Stumme, G.: Enhancing Social Interactions at Conferences. it - Information Technology **53**(3) (2011) 101–107
41. Goodman, B., Flaxman, S.: European Union Regulations On Algorithmic Decision-Making and a "Right to Explanation". arXiv:1606.08813 (2016)

42. Hoffmann, A., Söllner, M., Fehr, A., Hoffmann, H., Leimeister, J.M.: Towards an approach for developing socio-technical ubiquitous computing applications. In Heiß, H., Pepper, P., Schlingloff, H., Schneider, J., eds.: Informatik 2011. Volume P-192., Berlin, Germany, LNI (2011) 1–15
43. Behrenbruch, K., Atzmüller, M., Evers, C., Schmidt, L., Stumme, G., Geihs, K.: A Personality Based Design Approach Using Subgroup Discovery. In: 4th International Conference on Human-Centred Software Engineering. (In Druck)
44. Atzmueller, M., Baraki, H., Behrenbruch, K., Comes, D., Evers, C., Hoffmann, A., Hoffmann, H., Jandt, S., Kibanov, M., Kieselmann, O., Kniewel, R., König, I., Macek, B.E., Niemczyk, S., Scholz, C., Schuldt, M., Schulz, T., Skistims, H., Söllner, M., Voigtmann, C., Witsch, A., Zirfas, J.: Die VENUS-Entwicklungsmethode: Eine interdisziplinäre Methode für soziotechnische Softwaregestaltung. Technical report, Research Center for Information System Design (ITeG), University of Kassel (2014)
45. Roßnagel, A., Jandt, S., Geihs, K.: Socially compatible technology design. In David, K., Geihs, K., Leimeister, J.M., Roßnagel, A., Schmidt, L., Stumme, G., Wacker, A., eds.: Socio-technical Design of Ubiquitous Computing Systems. Springer International Publishing (2014) 175–190
46. Spiekermann, S.: User Control in Ubiquitous Computing: Design Alternatives and User Acceptance. Berichte aus der Wirtschaftsinformatik. Shaker (2007)
47. Luhmann, N.: Trust and Power. John Wiley and Sons Ltd. (1979)
48. Söllner, M., Hoffmann, A., Hoffmann, H., Leimeister, J.M.: How to use behavioral research insights on trust for hci system design. In: ACM SIGCHI Conference on Human Factors in Computing Systems (CHI), Austin, Texas, USA (2012)
49. Söllner, M., Hoffmann, A., Hoffmann, H., Wacker, A., Leimeister, J.M.: Understanding the formation of trust in it artifacts. In: Proceedings of the International Conference on Information Systems (ICIS), Orlando Florida, USA (2012)
50. Roßnagel, A.: Datenschutz in einem informatisierten Alltag. Studie für die Friedrich Ebert-Stiftung, Berlin (2007)
51. Brown, T., et al.: Design thinking. Harvard business review **86**(6) (2008) 84
52. Dorst, K.: The core of 'design thinking'and its application. Design studies **32**(6) (2011) 521–532
53. Schwaber, K.: Scrum development process. In: Business object design and implementation. Springer (1997) 117–134
54. Schwaber, K., Beedle, M.: Agile software development with Scrum. Volume 1. Prentice Hall Upper Saddle River (2002)
55. Lifschitz, V.: What Is Answer Set Programming? In: AAAI. (2008) 1594–1597
56. Guven, C., Atzmueller, M.: Applying Answer Set Programming for Knowledge-Based Link Prediction on Social Interaction Networks. Frontiers in Big Data (2019)
57. Atzmueller, M., Güven, C., Seipel, D.: Towards Generating Explanations for ASP-Based Link Analysis using Declarative Program Transformations, Cottbus, Germany, University of Cottbus
58. Etzioni, A., Etzioni, O.: Designing AI systems that obey our laws and values. CACM **59**(9) (2016) 29–31
59. van Otterlo, M.: From Algorithmic Black Boxes to Adaptive White Boxes: Declarative Decision-Theoretic Ethical Programs as Codes of Ethics. In: Proc. AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society. (2018)