SE4AI Issues on Social Media Agent Design with Use Cases

Extended Abstract

Francisco S. Marcondes
ALGORITMI/LASI, U. of Minho
Braga, Portugal
francisco.marcondes@algoritmi.uminho.pt

José João Almeida
ALGORITMI/LASI, U. of Minho
Braga, Portugal
jj@di.uminho.pt

Paulo Novais
ALGORITMI/LASI, U. of Minho
Braga, Portugal
pjon@di.uminho.pt

ABSTRACT

This paper is the result of an endeavor of specifying a social media agent through Use Case 2.0 (the "agile Use Case"). That what was expected to be a straightforward specification task revealed issues that subverts a critical foundation of the Use Case conception, nonexistent use-case between the SuD and the actor, yielding to the extensions proposed in this paper.

KEYWORDS

Online Social Media, Social Media Agents, SE4AI, Use Case 2.0

ACM Reference Format:
Francisco S. Marcondes, José João Almeida, and Paulo Novais. 2023. SE4AI Issues on Social Media Agent Design with Use Cases : Extended Abstract. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 3 pages.

1 INTRODUCTION

A Social Media Agent (SMA) is an agent that runs in an Online Social Media (OSM). They are built to achieve a varied range of purposes [4, 6, 18, 23, 25, 27]. The Use Case development process (UCP) is being successfully applied as Agent-Oriented Software Engineering (AOSE) approach in multi-agent system design [2, 3, 8, 13, 19, 32, 35] for decades, being then considered an industry standard [1]. Therefore, UCP is a natural choice. Yet, on applying it, unexpected issues emerged.

For a brief description, the UCP *cf.* [14, 16, 17, 31] follows the model-driven rationale, *i.e.* successive refinement upon models until the source-code can be delivered. The Use Case diagram (UCD) is responsible for depicting the system under development (SuD) behavior from an usage perspective through use-cases (small caps and hyphen for the oval symbol in the UCD). An actor models an event source that can be factored into roles, for realizing the persona's intention [30]. The interaction between the actor and an use-case is described by a Use Case narrative (UCN) whose developed collaborations results in the Use Case realization (UCR).

AOSE approaches that employ Use Cases such as ASPECS/PASSI [5], Prometheus [26], *etc.* are usually Unified Process inspired methods, therefore prescriptive and relying on Use Case 1.0. Therefore, subject to the same difficulties raised in this paper. In addition, since this proposal adopts UCP 2.0 (or "agile Use Case") [17] it does not run in unmanageable backlog such other user-story based agile AOSE such as [29]. Finally, unlike employing Use Cases just for requirements, it is applied as a whole development process.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

2 PROBLEM STATEMENT

Understating the big-picture is a software engineering principle, especially stressed in UCP [17]. The Use Case diagram (UCD) is the artifact responsible for depicting the system interface from the user perspective, then, the program's big-picture. In short, a UCD, is composed by an actor connected with a use-case with a rectangle denoting the system's boundary [17].

A difficult raise when defining the roles and actors. The Twitter User is presumably a role, but its link with the system under development (SuD) is multistable[1]. At the same time that the Twitter User interacts (*e.g.* tweet, like, *etc.*) with the SMA it does not do it directly but mediated by the OSM. Also, that interaction is collateral, not intentional, *i.e.* the Twitter User 's use-case is within the OSM boundary and not within the SuD's. This, however, is not a business[2] Use Case situation as the Twitter User does not aim to, specifically, interact with the SuD but is driven to it by, instancing, microtargeting strategies [22]. The SMA is the SuD, therefore it must be modeled *through* use-cases and *not have* use-cases. It is possible to consider the SuD as an OSM actor (*e.g.* a `Twitter` actor), but that would result in an inconsistent model as the use-case would be placed within the OSM boundary, but that's what is being built it the actor and not the OSM. In addition, considering the relation between the `Bot Master` and OSM would suggest that the `Bot Master` owns businesses use-cases with the SMA, that is also inconsistent by the same reason; the SMA is the SuD. However, it resembles a business use-case in the sense that the `Bot Master` 's use-cases is on a different abstraction level from the SMA's. For an instance, the `like` action, in isolation, does not deliver value to the `Bot Master` but it is certainly an aspect of the SMA's behavior.

The incorporation of all these concerns into a UCD would result in an awkward model, depicted for reference in figure 1. The problem to be tackled is how to set up the UCP for providing a suitable drive to develop SMA projects.

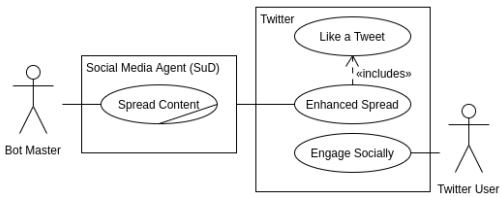

Figure 1: Illustration of the addressed problem.

[1] Multistability is a cognitive phenomenon on which the observer undergoes through spontaneous perceptual changes, *e.g.* alternates between a vase and two faces [21].
[2] A business Use-Case models an enterprise's business processes with the actors being entities external to the company, such as "customer" [15].

3 PROPOSED EXTENSIONS

The Use Case Environment. A first step towards addressing the proposed problem is to explicit the OSM as the environment [32, 34]. For doing so, it is necessary to introduce an additional element to the UCD for denoting the *environment* entity; to be called as Use Case environment (UCE) and stereotyped as «world». That informs, in the big picture, that the SuD and the Twitter User may interact through the world, with each one pursuing its own use-case.

On a general basis, the UCE can be realized through any convenient way. From a modularity perspective, Kripke world *cf.* [9] is a worthy choice. That suggestion, in short, is due to its accessibility relation, that enable an agent to run on an eventually unknown environment if that environment is accessible (*i.e.* if analogous properties hold). The properties, would be the interface with the world, the actions that the agent is capable of performing in it, constrained by the "laws of the Universe" *cf.* [7]. These laws, despite may include some parts of, are not the "service agreement", but the rules used by the mediation algorithm for boosting, de-boosting or banning a profile. Since the mediation algorithm rules are not publicly available, modeling it as a game *cf.* [12] is a good choice.

The Use Case Diagram. For improving the UCD presentation, consider that, formally, the SuD is a classifier and the use-cases its interfaces [11]. This implies that, the SuD has an optional responsibility slot [31]. That may receive the SMA's "use-case", organizing its narratives and encapsulating its realization. As the SuD is viewed as a distinctive type of class and the use-case as a distinctive type of interface, it is worth to keep along by calling the SMA's responsibility as *assignment* . The *assignment* then, is a degenerated use-case used for representing the agent's goals.

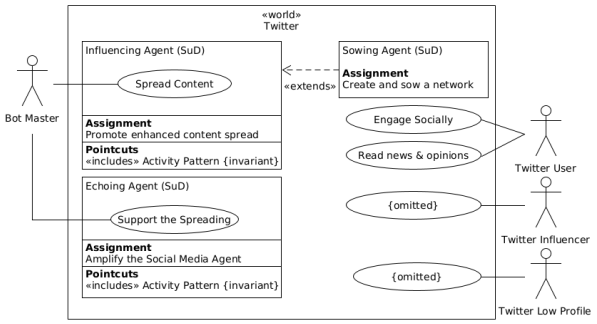

Figure 2: Extended Use Case diagram for `Twitter` .

The Use Case Narrative. The Use Case is based on the idea that value is delivered as the program is used, therefore it focuses on describing usage scenarios, called Use Case narrative (UCN), to drive the development [17]. The non-SuD use-cases are not worthy to be specified unless it helps the analysis procedure. For avoiding such a confusing model, it is worth to keep the UCN straight by only including steps that deliver value to the actor. The other steps would compose something analogous to the UCN but related with the *assignment* called *assignment* case narrative (ACN). The relation between an UCN and an ACN is of fulfillment as the *assignment* aims to accomplish the use-case's goal.

For illustration, see UCN with a `Create and Sow a Network` step

> (1) the `Bot Master` asks for the SMA to build a network based on a `target` ; the SuD start to engaging with them aiming turning them into followers
> **Exceptions**
> (a) the SMA could not create a list of candidates; the SuD ...

and the related ACN step

> **Related Play:** $A \xrightarrow{s} U \overset{f}{\underset{n}{<}} \begin{matrix}1\\-1\end{matrix}$
> (**Payoff-tree Legend**: *A* stands for SMA, *s* for profile setup, *U* for Twitter User ; *f* for a `follow` event; and ***n*** for a `nothing` happened event)
> (1) the `Bot Master` set a `target` , the SuD returns a resulting *reference profile list* for approval and modifications.
> (a) the SuD crawl the Twitter looking for candidates that meets the setted target. It then look into each candidade's followers aiming to retrieve features to be used for establishing following patterns.
> (2) ...
> **Variations**
> (a) if the `Bot Master` provides and `Twitter` credentials, the SuD ...

As told, the UCE and SuD relation benefits from being a game. Accordingly, the SMA and Twitter User is also a game, yet behavioral[3] [10]. The "UCE game" constraints the moves to be played in the "Twitter User game". Thence, the first is depicted as the use-case with the invariant clause in figure 2) and the second in the ACN. From a behavioral game perspective, the presented strategy is based on a simple mirroring approach for establishing rapport [10, 24].

The Use Case Realization. The Use Case realization (UCR) is used for providing a sharp separation between requirements and design [17]. It encompasses all model stack, including the analysis model, high-level design, low-level design and the implementation model [31]. The software product is a, hopefully traceable, elaboration of the UCR scattered through several components [16]. Remark that the UCR is build upon the agent (see figure 3).

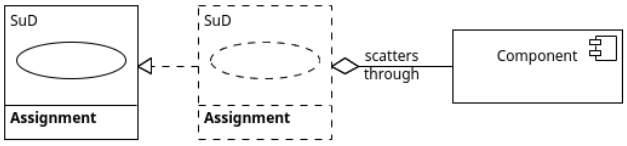

Figure 3: Relation between the UCD and UCR.

4 CONCLUSION

This paper stated the objective of presenting the difficulties found on developing a SMA and the extension proposal upon the Use Case 2.0 for addressing the presented issues. The difficulties can be summarized as the accidental relationship between the SMA and the Twitter User . The extensions, in turn, are specifying the environment where the agent works and a classifier for encapsulating its behavior in that world. A major concern was to keep these extensions compatible with the Use Case 2.0 proposal. Note that the examples refer to a SMA class to be instantiated as a swarm of coordinated agents, *e.g.* $10k \sim 350k$ [28, 33] (to be addressed in the process view of the 4+1 model [20]).

[3]Unlike game theory that assumes rational players [12], behavioral game theory include social utility, and other psychological issues as factors of decision-making [10].

ACKNOWLEDGMENTS

This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020

REFERENCES

[1] Reem Abdalla and Alok Mishra. 2021. Agent-Oriented Software Engineering Methodologies: Analysis and Future Directions. *Complexity* 2021 (2021).
[2] Nour Abdullatif and Sally Kassem. 2020. Modelling of Agent-Based Vehicle Routing Problem Using Unified Modelling Language. *Journal Européen des Systèmes Automatisés* 53, 6 (2020), 781–789.
[3] Mohammad MR Bhuiyan, MM Zahidul Islam, Aneesh Krishna, and Aditya Ghose. 2006. Co-evolution of agent oriented conceptual models and Use Case diagrams. In *2006 Sixth International Conference on Quality Software (QSIC'06)*. IEEE, 446–451.
[4] Eric Butow, Jenn Herman, Stephanie Liu, Amanda Robinson, and Mike Allton. 2020. *Ultimate Guide to Social Media Marketing*. Entrepreneur Press.
[5] Massimo Cossentino, Nicolas Gaud, Vincent Hilaire, Stéphane Galland, and Abderrafiâa Koukam. 2010. ASPECS: an agent-oriented software process for engineering complex systems: How to design agent societies under a holonic perspective. *Autonomous Agents and Multi-Agent Systems* 20 (2010), 260–304.
[6] Juan Echeverria and Shi Zhou. 2017. Discovery, retrieval, and analysis of the'star wars' botnet in Twitter. In *Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017*. 1–8.
[7] Jacques Ferber and Gerhard Weiss. 1999. *Multi-agent systems: an introduction to distributed artificial intelligence*. Vol. 1. Addison-wesley Reading.
[8] Peter Forbrig and Anke Dittmar. 2019. Stories, Use-Case Slices and Behavioral Models: Unifying Stakeholders' Views.. In *EICS Workshops*. 95–105.
[9] Dov M Gabbay, Franz Guenthner, et al. 1986. *Handbook of philosophical logic*. Vol. 3. Springer.
[10] Simon Gächter. 2004. Behavioral game theory. *Blackwell handbook of judgment and decision making* (2004), 485–503.
[11] Guy Genilloud and William F Frank. 2005. Use Case Concepts using a Clear, Consistent, Concise Ontology. *Journal of Object technology* 4, 6 (2005).
[12] Robert Gibbons et al. 1992. A primer in game theory. (1992).
[13] Carlos A Iglesias and Mercedes Garijo. 2005. The agent-oriented methodology MAS-CommonKADS. In *Agent-oriented methodologies*. IGI Global, 46–78.
[14] Ivar Jacobson. 1992. *Object-oriented software engineering: a use case driven approach*. Addison-Wesley Professional.
[15] Ivar Jacobson, Grady Booch, and James Rumbaugh. 1999. The unified process. *IEEE software* 16, 3 (1999), 96.
[16] Ivar Jacobson and Pan-Wei Ng. 2004. *Aspect-oriented software development with use cases (addison-wesley object technology series)*. Addison-Wesley Professional.
[17] Ivar Jacobson, Ian Spence, and Brian Kerr. 2016. Use-Case 2.0. *Queue* 14, 1 (2016), 94–123.
[18] Andreas Jungherr, Gonzalo Rivero, and Daniel Gayo-Avello. 2020. *Retooling politics: How digital media are shaping democracy*. Cambridge University Press.
[19] Elisabeth A Kendall, Margaret T Malkoun, and Chong H Jiang. 1996. A methodology for developing agent based systems for enterprise integration. In *Modelling and Methodologies for Enterprise Integration*. Springer, 333–344.
[20] Philippe B Kruchten. 1995. The 4+ 1 view model of architecture. *IEEE software* 12, 6 (1995), 42–50.
[21] Steven M Lehar. 2003. *The world in your head: A gestalt view of the mechanism of conscious experience*. Psychology Press.
[22] Jens Koed Madsen. 2019. *The psychology of micro-targeted election campaigns*. Springer.
[23] Steven Musil. 2019. *Cyberattack on Twitter targeted Epilepsy Foundation with strobing images*. https://www.cnet.com/tech/services-and-software/cyberattack-on-twitter-targeted-epilepsy-foundation-with-strobing-images/
[24] Shah-Abushakra Nasrine. 2021. Savvy Digital Citizenship: How to Master the Social Media World of Trolls, Bots, and Propaganda. online.
[25] Carl Öhman, Robert Gorwa, and Luciano Floridi. 2019. Prayer-bots and religious worship on twitter: A call for a wider research agenda. *Minds and machines* 29, 2 (2019), 331–338.
[26] Lin Padgham and Michael Winikoff. 2002. Prometheus: A methodology for developing intelligent agents. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*. 37–38.
[27] Peter Pomerantsev. 2019. *This is not propaganda: Adventures in the war against reality*. PublicAffairs.
[28] CGJ Putman, Lambert JM Nieuwenhuis, et al. 2018. Business model of a botnet. In *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*. IEEE, 441–445.
[29] Sebastian Rodriguez, John Thangarajah, and Michael Winikoff. 2021. User and System Stories: an agile approach for managing requirements in AOSE. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. 1064–1072.
[30] Doug Rosenberg and Kendall Scott. 1999. *Use case driven object modeling with UML*. Springer.
[31] J. Rumbaugh, I. Jacobson, and G. Booch. 2004. *Unified Modeling Language Ref. Man.* Pearson.
[32] Leon Sterling and Kuldar Taveter. 2009. *The art of agent-oriented modeling*. MIT press.
[33] Polly Wainwright and Houssain Kettani. 2019. An analysis of botnet models. In *Proceedings of the 2019 3rd International Conference on Compute and Data Analysis*. 116–121.
[34] Danny Weyns, Andrea Omicini, and James Odell. 2007. Environment as a first class abstraction in multiagent systems. *Autonomous agents and multi-agent systems* 14 (2007), 5–30.
[35] Haitao Zhang, Guiafng Wu, and Wenshao Bu. 2019. Research of Agent Based Control Model for Embedded Networked System. *Engineering Letters* 27, 2 (2019).