Notes on Daniel López TFG’s proposal: reputation-based models for Amara

* Title ES: Explorando potencialidades de las tecnologías descentralizadas en la colaboración abierta distribuida: plataforma blockchain para sistemas de reputación
* Title EN: Exploring the potential of decentralised technologies in crowdsourcing: a blockchain-based platform for reputation systems
* Abstract (ES):

El *crowdsourcing* - o colaboración abierta distribuida - es un fenómeno creciente en el que grupos numerosos de personas se coordinan en una convocatoria abierta a través de una plataforma para contribuir información, conocimiento o habilidades. Dicha colaboración se organiza típicamente alrededor de micro-tareas.

Esta propuesta de TFG, que se enmarca dentro de la investigación llevada a cabo en el proyecto europeo ERC P2PModels (<https://p2pmodels.eu>), explorará las potencialidades de las tecnologías descentralizadas (ej.: blockchain) con respecto al desarrollo de sistemas de reputación en el contexto de dichas plataformas crowdsourcing. Concretamente, se implementará un sistema de insignias (badges) que los participantes obtendrán en función de la cantidad y la calidad (evaluada por pares) de las micro-tareas realizadas. Dicho sistema usará tecnologías web (Javascript, React) y descentralizadas (Ethereum, Solidity) punteras. El código desarrollado será software libre, y existirá la posibilidad de publicar un artículo de investigación al concluir el mismo, si existe interés por parte del estudiante.

Los requisitos de dicho sistema, que serán guiados por el director, emergen de diversos talleres de co-diseño realizados mano a mano con participantes del proyecto Amara (<https://amara.org>). Amara es una plataforma de *crowdsourcing* para la creación de subtítulos utilizada por organizaciones como Khan Academy ([https://www.khanacademy.org](https://www.khanacademy.org/)), Scientific American ([https://www.scientificamerican.com](https://www.scientificamerican.com/)) o la California Academy of Science ([https://www.calacademy.org](https://www.calacademy.org/)). De esta manera, el TFG ofrecerá al estudiante la posibilidad de formar parte de un proceso de co-diseño con un caso de estudio real y paradigmático en el previamente citado contexto de crowdsourcing.

Preliminary requirements:

* Workers can earn absolute and temporary badges.
  + Absolute badges:
    - They are earned after having accumulated a set of actions, and the linguist keeps them in any case.
    - Examples of these badges include:
      * A badge earned after having captioned/translated X minutes. Ideally, these milestones could be configured, but some initial values could be X = 100, 200, 500, and 1000.
      * A badge earned after having reviewed Y minutes. Ideally, these milestones could be configured, but some initial values could be Y = 100, 200, 500, and 1000.
  + Temporary badges:
    - They consider the metrics during a period of time P (ideally, this should be configurable, but one month can be assumed to simplify the model if needed).
    - The badge is maintained as far as that rate is over a threshold T (ideally, this could be configurable, but possible ranges are 50%, 75%, and 100%).
    - Examples of these badges include:
      * A badge concerning the % of tasks captioned/translated on time over the period P.
      * A badge related to the feedback received by peers. The criteria for the evaluation, which another worker performs, should consider that these criteria might change over time. As a result, it should be designed as dynamic as possible. However, in order to simplify, during this stage, it could include:
        + The overall quality of the translation/caption: scale 1-5
        + Use of glossary: scale 1-5
        + Punctuation: scale 1-5
* For this simple prototype, the reputation in itself is still not considered in the model in order to provide more or less priority regarding the possibility of a task to be offered to a linguist L. During this first stage, this badge system could be integrated, for example, as a variant of the previously described Round-Robin model.
* A simple and experimental action suggested by the workers, which could be triggered because of having a low reputation and could be implemented at this point, is to receive a message (e.g. email, message on the platform) in which the organisation will provide options for training in order to improve the linguist's skills and, as a result, his/her reputation.
* The scope of the badges and any other elements of the worker's profile regarding his/her reputation should be chosen by each worker. The levels should include at least the following: totally private, visible to other platform members, and totally public. The default value would be private.
* The current rule of no more than one task at a time remains: no more tasks are offered to that linguist L until the task has been completed.
* The system should include an "onboarding subsystem” to clearly explain how each metric is calculated, how badges can be earned/lost, etc.

Technologies involved:

* react.js (particulary hooks)
* docker
* the graph
* ethereum
* solidity
* hardhat
* bash
* cron
* git
* github

Tecnical resources:

* Compilation of resources by P2PModels: <https://github.com/P2PModels/se-docs>
* Ethereum
  + <https://docs.soliditylang.org/en/latest/> (Documentación oficial de Solidity)
  + Books
* C. Dannen, Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners. Apress, 2017.
* W.-M. Lee, Beginning Ethereum Smart Contract Programming. Apress, 2019.
* M. Mukhopadhyay, Ethereum Smart Contract Development. Packt publishing, 2018
* React & JS:
  + <https://www.academia.edu/38744221/Alex_Banks_and_Eve_Porcello_Learning_React_EN>
* Bootcamps (suggested by Hanna Fiegenbaum)
  + Interchain Foundation -> Interchain Academy (<https://academy.cosmos.network/> )

Case study related resources:

* About the case study -<https://p2pmodels.eu/amara/>
* Paper on alternative models: <https://dl.acm.org/doi/10.1145/3479986.3479987>
* Technical documentation (ongoing): <https://task-allocation.docs.p2pmodels.eu/>

Example of technical outcomes for a different case study (Smart):

* Backend (including smart contracts):<https://github.com/P2PModels/wallofshame-backend>
* Frontend:<https://github.com/P2PModels/wallofshame-frontend>
* Deployed version:<https://observatorio.p2pmodels.eu/>

Other resources:

* Guidelines TFG <https://informatica.ucm.es/tfgs-2022-2023>