Formales Modell einer Domänenspezifische Sprache zur verteilten Programmierung mit Mehrheitenbildung

Im Internet werden zunehmend Inhalte kollaborativ erzeugt. Dabei entsteht ein Ergebnis durch Beiträge einzelner **Akteure**. Eine zentrale Frage ist die Bewertung der einzelnen Beiträge, und damit ihren Anteil am Gesamtergebnis.

Wikipedia folgt einem streng hierarchischem Modell, in welchem Vertrauenspersonen die Ihnalte der Benutzer filtern. Die Verantwortung liegt bei der Organisation. Effizienter sind jedoch selbstregulierende Systeme, bei denen die Benutzer die Inhalte der Anderen bewerten. Beispiele wären die auf *Voting* und *Reputation* basierenden Plattformen Reddit[1] und StackOverflow[2]. Ein solches Prinzip könnte man auf das Verwalten aller digital geteilter Inhalte verallgemeinern.

Nehmen wir z.B. an, es gäbe eine Webseite, die sich im Besitz von Akteuren befindet. Alle Akteure wollen **gerecht**, also proportional zu ihrem Besitz, über die Inhalte der Webseite entscheiden können, sowie ihre Entscheidungsgewalt in bestimmten Bereichen an andere vertrauenswürdige Akteure delegieren können. Dieses gilt für die medialen Inhalte, die Programmierung, die Architektur, die Wertflüsse wie ein Geteiltes Budget oder eine Einkommensverteilung, sowie nicht automatisierbare Prozesse, wie das Validieren neuer Beiträge. Das eigentliche Ergebnis wird anhand von einer Mehrheit der Besitzer bestimmt.

Die Piratenpartei hat mit *Liquid Democracy*[3] einen deligierten Abstimmungsprozess digital abgebildet. Jedoch bedarf es bei ihrem Ansatz eines zentralisierten Servers, welcher als Angriffspunkt die Sicherheit des Prozesses gefährdet, da die Akteure auf die Korrektheit des Servers vertrauen müssen.

Auch eignet sich das Konzept nur bedingt um damit geteiltes Eigentum wie z.B. eine Webseite zu modellieren, da der Besitz und somit die Stimmengewichtung nicht unter den Mitgliedern gleichverteilt ist. Auch fehlt es an eienr Rechte- und Rollenverteilung.

Das 2009 eingeführte Konzept des Bitcoins[4] hat zumindest den Aspekt der Sicherheit gelöst, indem sie die Verteilung von Tokens unter Akteuren auf eine dezentrale Weise modellieren. Die Manipulation der Tokens ist durch eine Identität und ein kryptografisches System **gesichert**.

Sicher bedeutet hierbei, dass einmal getroffene Vereinbarungen auch eingehalten werden.

Das Ethereum Team hat die grundlegende Technologie des Bitcoins generalisiert[5].

[1] http://reddit.com

[2] http://stackoverflow.com

[3] Friedrich Lindenberg. Konzeption und Erprobung einer Liquid Democracy Plattform anhand von Gruppendiskussionen. TU Ilmenau, 2010.

[4] Satoshi Nakamoto, Bitcoin: A peer-to-peer electronic cash system. http://bitcoin.org/bitcoin.pdf

[5] Gavin Wood, ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER. http://gavwood.com/paper.pdf

Sie haben eine virtuelle Maschine mit einer turing vollständigen Maschinensprache entwickelt. Diese erlaubt das *sichere* Ausführen von Berechnungen in einem dezentralen Netzwerk, welches nicht nur einen Informationsfluss, sondern auch durch die Einbindung der lokalen Währung Ether, einen Wertefluss ermöglicht.

Es entsteht eine Vielzahl von neuen Anwendungsmöglichkeiten: verbindliche, autonome Verträge zwischen mehreren Parteien, dezentrale autonome Organisationen (DAO) oder profitorientierte Kooperationen (DAC).

Ein Beispiel einer solchen DAO ist das *namecoin* Konzept, welches als Alternative zur ICANN Organisation die TLD ".bit" verwaltet und in naher Zukunft die ICANN ablösen könnte.[6] Die Regeln unter denen DACs und DAOs funktionieren, wie das Bewilligen einer neuen TLD, werden auf der Ethereum VM programmiert. Die Einigung der Akteure auf ein Programmstand geschieht noch durch eine Vertrauensinstanz, z.b. bestimmten Schlüsselpersonen.

In dieser Arbeit möchte ich ein Modell einer Meta-Programmiersprache für verteiltes Programmieren auf Basis von Ehtereum entwickeln und untersuchen. In dieser werden Einigungsprozesse als Bestandteil des Entwicklungsprozesses angesehen.

Grundlegende Aktionen eines Akteurs ist das Vorschlagen von Alternativknoten (klassisches Programmieren), sowie das Partizipieren an Wahlen über Alternativen.

Das Modell soll mit formaler Logik sowie modelltheoretischen Konzepten beschrieben werden und anschließend auf Machbarkeit und Wiederspruchsfreiheit untersucht werden.

Der Prozess des "transitive delegated voting" aus Liquid Democraty dient als Grundlage des Wahlprozesses. Diese wird durch das Modell von Eigentum und Delegationsbedingungen erweitert, welches eine Rechte- und Rollenverwaltung ermöglicht. Eine Mehrheit der Besitzer bestimmt dabei die für die Programmausführung gewählte Optionen.

Die Syntax der Programmiersprache ist ein LISP Dialekt, mit dem Paradigma, dass Knoten des abstrakten Syntaxbaumes als Daten sowie als Code interpretiert werden können. Die Daten befinden sich teilweise in einem kryptografisch gesicherten, verteilten Netzwerk ähnlich der auf BitTorrent aufbauenden IPFS[7] oder der Maidsafe[8] Architektur und teilweise in dem Ethereum Speicher.

Diese Sprache soll das oben beschriebene Problem der *gerechten* und *sicheren* Verwaltung der geteilten Webseite lösen.

---

[6] ICANN, Identifier Technology Innovation Panel - Draft Report. https://www.icann.org/en/system/files/files/report-21feb14-en.pdf

[7] Juan Benet, IPFS - Content Addressed, Versioned, P2P File System. http://static.benet.ai/t/ipfs.pdf

[8] David Irvine, MaidSafe Distributed File System. http://maidsafe.net/Whitepapers/pdf/MaidSafeDistributedFileSystem.pdf