Formales Modell einer Domänenspezifischen Sprache zur verteilten Programmierung mit Mehrheitenbildung

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

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.

Wikipedia folgt einem streng hierarchischem Modell, in welchem Vertrauenspersonen die Inhalte 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 weiteres Beispiel für die Bewertung von Inhalten ist das Konzept *Liquid Democracy*[3], ein Vorschlag der Piratenpartei für eine moderne politische Konsensbildung. Ein solches Prinzip könnte man auf das Verwalten aller digital geteilter Inhalte verallgemeinern.

Jedoch eignen sich diese Konzepte nur bedingt um damit geteiltes Eigentum wie z.B. eine Webseite zu modellieren, da Abstimmungen auf eine informale Weise vorgenommen werden. Es fehlt einer formalen Syntax, welche die Implikationen einer potentiellen Entscheidung vor der Wahl durch Simulation klarer zeigt, sowie nach der Wahl automatisch ausführt. Außerdem bedarf es bei den Ansätzen eines zentralisierten Servers, welcher als Angriffspunkt die Glaubwürdigkeit des Prozesses gefährdet, da die Akteure auf die Korrektheit des Servers vertrauen müssen.

Das 2009 eingeführte Konzept des Bitcoins und der Blockchain[4] bietet eine Alternative zur zentralen Server-Architektur. Dieses beschreibt ein Protokoll in einem Netzwerk, welches Inhalte aus einem gebildeten Konsens bereitstellt. Es besitzt eine einfache, nicht turing vollständige Skriptsprache, sowie durch ein asymmetrisches Kryptosystem, Rechte und Rollen. Neue Inhalte werden nach einer Validierung ebenfalls in den Konsens aufgenommen. Die einfachste Interpretation von Bitcoin ist die eines Werteträgers, wobei die Beschränkung der Skriptsprache wenig Spielraum für weitere Interpretationen lässt. Allge-

[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

meiner ist das auf dem Bitcoin-Protokoll aufbauende Ethereum [5], welches eine turing vollständige Skriptsprache besitzt. Es entsteht eine Vielzahl von neuen Anwendungsmöglichkeiten: verbindliche, autonome Verträge zwischen mehreren Parteien, dezentrale autonome Organisationen (DAO) oder profitorientierte Kooperationen (DAC), die allesamt Werte- und Informationsflüsse ermöglichen. Ein Beispiel einer solchen DAO ist das *namecoin* Konzept, welches als Alternative zur ICANN[6] Organisation die TLD ".bit" verwaltet und in naher Zukunft die ICANN ablösen könnte.[7] 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 zentrale 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.

Die Grundlegende Idee ist, das der Besitz eines Programms eindeutig beteiligten Akteuren zugeordnet ist. Aktionen eines Akteurs ist das Vorschlagen von Alternativknoten (klassisches Programmieren), sowie das Partizipieren an Wahlen über Alternativen, für welches "transitive delegated voting" aus Liquid Democracy als Grundlage verwendet wird.

Technisch werde ich eine Funktion beschreiben, die eine gegebene eindeutige kontextfreie Grammatik so erweitert, dass diese zu der gesuchten verteilten Grammatik wird (im Folgenden **SCFG** genannt). Worte aus $L(SCFG)$ beinhalten die Besitzverteilung beteiligter Akteure, sowie Optionsmengen, Delegierungen und Stimmen für Optionen. Eine Konsens-Funktion erzeugt aus einem Wort der $L(SCFG)$ Sprache ein Wort der ursprünglichen Sprache ($L(G)$). Manipulation eines Wortes der $L(SCFG)$ Sprache sind Regeln unterworfen, hierdurch können die Akteure nur ihre eigenen Stimmen und Delegationen bearbeiten. Optionsmengen können an beliebiger Stelle erstellt und erweitert werden, solange ein Wort mit der neuen Option ebenfalls ein valides Wort aus $L(G)$ bildet. Die Manipulation des Wortes aus $L(SCFG)$ wird durch ein Transitionssystem modelliert. Durch die verifikation des Ethereum Netzwerkes ist eine dezentrale Interaktion der Akteure möglich.

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

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

[6] Internet Corporation for Assigned Names and Numbers

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