ショートノート

Research Notes

インタラクション設計言語 Q の提案

Interaction Design Language Q: The Initial Proposal

石田 亨
Toru Ishida
京都大学情報学研究科 社会情報学専攻 & 科学技術振興事業団 (CREST)
Graduate School of Informatics , Kyoto University Japan & Science and Technology Corporation (CREST)
ishida@i.kyoto-u.ac.jp,http://www.lab7.kuis.kyoto-u.ac.jp/

福本 理人
Masahito Fukumoto
科学技術振興事業団 (CREST)
Japan Science and Technology Corporation (CREST)
fukumoto@digialcity.jst.go.jp

keywords: autonomous agent, agent communication language , multi-agent systems

Summary

We started working on Q, an interaction design language, which is to describe and experiment interaction among autonomous agents and humans. Unlike previous agent communication languages, Q is intended to design various interaction patterns without depending any internal models of agents. Unlike previous protocol description languages, Q cannot guarantee the correctness of protocols. We rather combine execution and planning layers to create robust behaviors of agents.

1. はじめに

従来のエージェント記述言語は, エージェントの内部メカニズムのモデル化を目的にしてきた. これは AI が, 知的な処理の計算モデルを追求するものであるところからきている. 例えばプロダクションシステムや様相論理は, いずれもエージェントの内部メカニズムを説明するためのモデルであるが, 膨大な内部処理を必要とするエージェントを実装するのは現実的ではない. 今日, エージェント記述言語が使われていない原因は, エージェントの内部メカニズムを説明するための科学的モデルを, そのまま実装に用いようとしたためではないだろうか.

ここで提案する Q は, インタラクション設計言語である. インタラクション設計言語は, エージェントの内部メカニズムの記述を目的としたものではない. エージェント（あるいはエージェント群）に対し, 人間（あるいはエージェント) が外部からどのように作用するかを表すものである. シナリオの記述をエージェント群に与えることによって, エージェント群とのインタラクションを設計する（エージェント群にそのように振舞うことを依頼する）ものである.

エージェント記述言語からインタラクション設計言語への視点の移行は, 言語仕様に対し, また言語処理系に対し, 大きな変化を生じさせる. Q は, エージェントがどのような言語を用いてどのように記述されているかに感知しない. 例えば, エージェントが外部から「走れ」「歩け」という 2 つの依頼しか受け付けないのであれば, Q の言語仕様は「走れ」「歩け」の 2 つの構文を許すだけである. さらにその構文の意味するところは, 実際にエージェントに対して依頼を発してみなければ分からない. 「走れ」という依頼を与えた際に, 大きく手を振って速く走るか, 軽快にジョギングするかはエージェントによるのであって, その実装は Q の記述には影響しない.

シナリオライターは Q を用いて, 複数のエージェントに対する依頼を記述する. 従来からのプロトコル記述言語との相違は, その依頼が, 依頼を受けるエージェントの全ての行動を規定するものではなく, シナリオライターの観点からの部分的な記述にすぎないことである. 実際, エージェントは, 複数のシナリオを異なる依頼元から同時に受け取ることがある. それらが矛盾していれば, 全てを実行することはできない. よく設計されたエージェントは, 複数のシナリオを整合させるよう試み, それが不可能と知れば依頼元に調整を求めるだろう.

また, シナリオライターは訓練されたプロトコル設計者ではなく, エージェントシステムを利用する応用プログラマ（あるいはエンドユーザ）であることも本質的な違いである. Q 言語で書かれたシナリオには誤りが存在することを覚悟しなければならない. エージェントは, 誤りのないシナリオを期待するのでなく, 誤りを含むシナリオを頑健に実行することを要求される.

2. インタラクションの設計

Q は母言語として Scheme を用いている. シナリオの記述のために, 観測（キューと呼ぶ）と動作（アクション

と呼ぶ）を記述する特殊形式と, ガード付きコマンドが導入されている．Q の母言語を Scheme としたのは, プログラムをデータとして扱える Lisp 系言語の特徴がインタラクションの記述に適しているからである．例えば, A に「明日, B の仕事がもし早めに終わったら, 『これこれしかじか』の仕事を頼んでください」と依頼する場合を考えよう．『これこれしかじか』はシナリオを表すが, そのシナリオはデータとして A から B に送信され, そこで初めて解釈実行される．プログラムをデータとして扱えることが, 依頼を表すシナリオの記述には必要なのである．以下, 基本的な言語機能を説明する．

(1) キュー

インタラクションのきっかけはキュー (合図あるいは手がかりと訳されることが多い) と呼ばれる．Q におけるキューは外界を観測するだけで, 外界への副作用はない．例えば以下のように記述する．

```
(?hear “Hello” :from Hanako)
(?see Station :direction “South”)
```

(2) アクション

Q でのアクションは外界への作用である．アクションは, 以下のように記述する．

```
(!move self :from Station :to Bus_Terminal)
(!speak “Hello” :to Hanako)
```

キューやアクションの記述はエージェントによって解釈され実行されるので, 同じアクションでも, エージェントごとに意味するところが異なってよい．しかし, 複数のエージェントで共通に用いられる単語（例えば, hear, see, speak）は, 可読性を高めるために Q によって予約され, 属性などが統一的に与えられている．

(3) ガード付きコマンド

多くのキューを並行して探索する, あるいは待ち受けるには, ガード付きコマンドを用いる．

```
(guard ((?hear “Hello” :from Hanako)
        (!speak “Hello” :to Hanako))
       ((?see Station :direction “South”)
        (!move :from Station :to Bus_Terminal))
       (otherwise
        (!send “I am waiting” :to Taro)))
```

この場合には, いずれかのキューが観測された後に, 後続する形式が実行される．どのキューも成立しない場合には, otherwise 節が実行される．この他, Q 言語では, Scheme が提供する様々な制御構文 (条件分岐, 繰り返し) が利用可能である．

(4) シナリオ

シナリオは状態遷移を記述するためのものである．

```
(defscenario chat (message)
  (let (($x #f))
    (scene1 ((?hear“Hi!”:from $x) (go scene2))
            ((?hear “Bye”) (go scene3)))
    (scene2 ((equal $x Taro) (!say message))
            (otherwise (!say “Hello”))
    (scene3 (#t (!say “Bye”)))))
```

上記の例では, 3 状態 (scene1-scene3) が定義されている．各状態の記述はガード付きコマンドと同様の構文則に従う．各状態から任意の形式（関数やシナリオを含む）が呼び出せるので, 柔軟にエージェントの行動を定義することができる．

Q を用いたシナリオ記述は以下のように進められる．まずシナリオライター（計算機科学の非専門家であることが多い）とエージェントシステム開発者（計算機科学の専門家）が協議し, キューとアクションを両者のインタフェースとして定める．従って Q には, エージェントの動作を詳細に記述する「JAVA 言語呼び出し」のような, 直接実行を想定した機能は存在しない．次にシナリオライターは定められたキューとアクションを用いて, 上記の Q の構文を使いながらシナリオを記述する．一方, エージェントシステム開発者は, 協議の結果のキューとアクションを実装する（実際には Q インタフェースのための簡単なプログラムを追加するだけでよいことが多い）．エージェントシステムの開発者とユーザのインタフェースを規定できることが, Q を導入する最大の効果である．

3. 解 釈 実 行 系

Q で記述されたシナリオは, Q メッセンジャーと呼ばれる解釈実行系によって処理される．Q メッセンジャーは各々のエージェントシステムに対して生成される．エージェントシステムは 複数のエージェントをその中に含むことができる．Q メッセンジャーのプログラムインタフェースは, Scheme, C++, JAVA に対して用意され, 以下の 2 レイヤからなる．

- 実行レイヤ
- プラニングレイヤ

実行レイヤでは, Q インタプリタはエージェントシステムに, 各々のエージェントが次に実行すべきキューとアクションを提示する．シナリオの実行は Q インタプリタが担当し, キューとアクションの実行はエージェントが受け持つ．

シナリオを実行するだけであれば, 実行レイヤだけで十分である．しかし, 非専門家により記述されたシナリオには誤りが含まれるのを防ぐことはできない．さらに非専門家は仕様を記述しないため, シナリオの検証は不可能である．Q では人間社会と同様のプロセスで, 誤りが除去されていく．シナリオライターはエージェントと対話することによって, 動作を修正していく．例えば, 壁にぶ

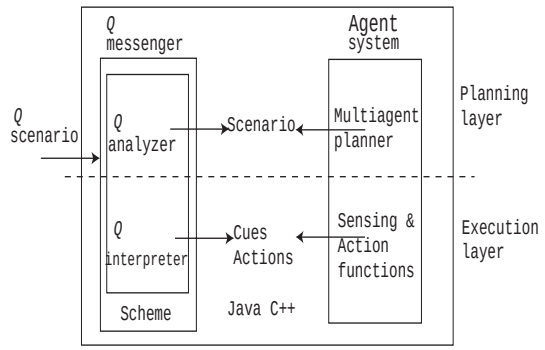

図 1 Q 解釈実行系

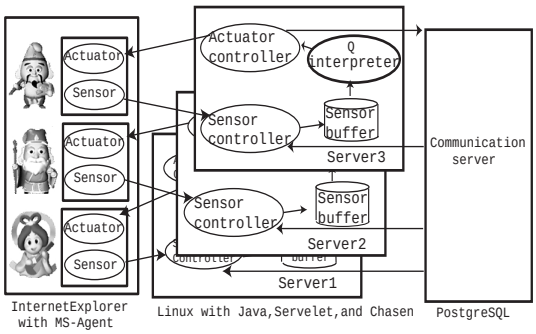

図 2 Venus&Mars のシステム構成図

つかって動けなくなったエージェントがいると、シナリオライターは「何をやってるの？」と声をかける．エージェントは「このシナリオを実行しているんです」とシナリオの一部を示す．シナリオライターは「ごめん, そうじゃなかった．シナリオをこう修正してくれる？」と依頼する．我々はこれを「社会的デバッガ」あるいは「劇場型デバッガ」と呼んでいる．

エージェントがシナリオに問題を発見したような場合には, 処理はプラニングレイヤに移行する．プラニングレイヤでは, Q アナライザがシナリオをソースレベルで解析する．その結果を用いて, エージェントはシナリオライターと対話を開始しシナリオの修正を求める．

エージェントシステムは多数のエージェントを含むことができる．また, 系全体には多数のエージェントシステムが存在してよい．後述する仮想空間システム FreeWalk では, 一つのエージェントシステム内で 1000 体のエージェントが並行に動作する．これを制御するためには, Q インタプリタが並行に 1000 のシナリオを解釈実行しなければならない．こうしたマルチエージェントのシナリオ実行で問題となるのは, その並行性のために実行毎に結果が異なりうることである．これを防ぐには, 並行処理制御を OS に委ねずに, インタプリタで制御する必要がある．Q の母言語として Scheme を選択したもう一つの理由は継続（continuation）を利用して擬似的に, Q インタプリタの下で並行処理を実現できるからである．

4. 応　　用

Q は, 複数の先進的応用研究の要求を満たす形で設計された．全く異なる応用システムが Q を共有できるのは, Q がエージェントの内部を記述するためのものでなく, ユーザの視点からエージェントにシナリオを依頼するための言語だからである．

(1) マルチエージェント情報検索

WEB 検索支援システム Venus&Mars (イメージ情報科学研究所) では, 複数のエージェントが独立に WEB の検索を行う [Kitamura 01]．通常の方式であれば, システム内に黒板を設け, 各エージェントの検索結果を黒板上で調整する．しかし, 複数の検索エージェントが独立に開発された場合には, 調整結果は満足なものとならない場合が多く, ユーザからの信頼は得られにくい．Venus&Mars は検索エージェントの調整過程を, 複数のキャラクタの対話の形でユーザに見せてしまうというユニークな発想に基づいている．ユーザがエージェント間の協調過程を観察でき, さらにその過程に参加できれば, システム全体への信頼感が増すと考えたのである．例えば図 2 では, 画面上に 3 エージェントが登場し, 連携することで WEB 検索を実現している．JAVA で記述された複数の検索エージェントそれぞれに Q メッセンジャーが用意され, 与えられた Q シナリオを実行している．

(2) 3 次元仮想空間での避難シミュレーション

図 3 は, デジタルシティプロジェクト（科学技術振興事業団）の仮想空間システム FreeWalk [Nakanishi 99] での避難シミュレーションを示している．仮想空間内では C++で記述された多数のエージェントがシナリオに基づき行動する．実証実験のカバーストリーは以下のとおりである．

> 200X 年, 京都の 3 次元仮想都市 [Ishida 02] で, 京都駅, 地下鉄での災害を想定した避難訓練が実施され, インターネット経由で市民多数が参加する．
>
> 200Y 年, 京都で地震が発生．火災が地下鉄, 京都駅構内で発生する．物理空間での状況はセンサから無線ネットワークを通じて刻々センターに送られる．センターでは人々の群れとしての行動が把握されパネルに表示される．さらに仮想都市での避難訓練の結果を利用して, 適切な指示がモバイル端末に送られ, エージェントによる誘導が行われる．

実証実験の第一段階では, インターネットから参加する 100 人のアバターと, 1000 体のエージェントを仮想京都駅で動作させ, 避難シミュレーションを実施する．この段階でのエージェント群の制御は, 1 個の Q メッセンジャーで実現することができる．しかし, 第二段階になる

図 3 仮想空間 (四条河原町) でのシミュレーション

と, モバイル端末を含めエージェント群は分散して配置されることになり, その間の通信帯域も多様となる. こうした環境で効率よくシナリオを実行することは今後の課題である.

(3) エージェントの社会心理学実験

FreeWalk と Q を用いてエージェントの性質を調べるための社会心理学実験を進めている [Isbister 00]. この実験では, 計算機科学の非専門家（社会心理学者）が, 様々に設定を変えて繰り返し実験を行うことが必要となる. 現在, Stanford 大学の社会心理学コース（複数の学生グループがそれぞれ実験を計画し実施する）に FreeWalk と Q を提供する計画を進めている. シナリオ記述が, エージェントの開発者とユーザのインタフェースとなる典型的な例と考えている.

5. お わ り に

将来, 物理空間と仮想空間が連動する環境が広がれば, 様々な社会指向サービスの実現のために, 人間とソフトウェア/ハードウェアエージェントとの多彩なインタラクションを記述する必要が生じる. このとき各々のエージェントが独立に開発されたものであれば, アドホックな協調系は不安定なものとなるだろう. エージェントに対して, いかに振舞えばよいかという社会的制約をシナリオとして与えることが必要となる. また, その記述は, 計算機科学の非専門家によって与えることができるものでなければならない.

今後の５年間に, Q の仕様を発展させその処理系を開発する. 例えば「押されると倒れる」という現象は, 本論文の範囲では記述できない. 行為者（エージェント）の視点からのシナリオとは別に, 行為の対象（オブジェクト）の性質の記述が必要となるからである. エージェントとオブジェクトが明らかに違うものとして意識されるのも,「依頼」という抽象度の高いレベルでの記述を考えているためである. さらに, 記述されたシナリオ（社会的制約）の下で, 環境に適応して行動を選択するエージェント（社会的エージェントと呼ぶ）の動作メカニズムが研究課題となる. この研究は, 人とエージェントのインタラクションを記述するという点で, これまでにない研究の方向性を人工知能分野に持ち込むことになる. エージェントにとっては, シナリオという社会的制約の下での自律性が要求されるのである.

本研究の社会に対する貢献は, 都市におけるナビゲーションなどの様々なサービスの記述や, 大規模な社会シミュレーションが記述可能となることである. さらに, 社会心理学者や防災関係者などの計算機科学の非専門家と, エージェントの開発に携わる計算機科学の専門家とのインタフェースを明確な形で記述でき, 研究分野間のコラボレーションに寄与できると考えている.

謝　　辞

大阪市立大学の北村泰彦助教授, 京都大学の中西英之助手, NTT サイバースペース研究所筒口けん博士, JST デジタルシティ研究センター, イメージ情報科学研究所, 株式会社 CRC 総合研究所, 株式会社数理システムに様々な御協力を頂いたことを記して感謝します.

◇ 参　考　文　献 ◇

[Isbister 00] Isbister,K.,Nakanishi,H.,Ishida,T.,and Nass,C.:Helper Agent: Designing an Assistant for Human-Human Interaction in a Virtual Meeting Space, CHI-00, pp. 57-64, 2000.

[Ishida 02] Ishida,T.: Digital City Kyoto: Social Information Infrastructure for Everyday Life,Communications of the ACM (CACM), 2002 (to appear).

[Kitamura 01] Kitamura,Y.,Yamada,T., Kokubo,T., Mawarimichi,Y., Yamamoto, T.,and Ishida,T.:Interactive Integration of Information Agents on the Web, M. Klusch, F. Zambonelli (Eds.), Cooperative Information Agents V, Springer-Verlag, pp. 1-13, 2001.

[Nakanishi 99] Nakanishi,H., Yoshida,C., Nishimura,T.,and Ishida,T.:FreeWalk: A 3D Virtual Space for Casual Meetings,IEEE Multimedia, Vol.6, No.2, pp.20-28, 1999.

〔担当委員：藤本和則〕

2001 年 10 月 16 日　受理

著　者　紹　介

石田　　亨(正会員)

1976 年京都大学工学部情報工学科卒業, 1978 年同大学院修士課程修了. 同年日本電信電話公社電気通信研究所入所. 米国コロンビア大学計算機科学科客員研究員, ミュンヘン工科大学客員教授, パリ第六大学招聘教授など. 現在, 京都大学大学院情報学研究科社会情報学専攻教授. 工学博士. IEEE Fellow. 人工知能, コミュニケーション, 社会情報システムに興味を持つ.

福本　理人

2001 年京都大学工学部情報学科卒業. 現在, 科学技術振興事業団に所属. 研究プロジェクト「デジタルシティのユニバーサルデザイン」においてインタラクション設計言語 Q の開発に従事.