Multiagent gaming system for multilingual communication

Yuu NAKAJIMA
Department of Information Science
Toho University
Chiba, Japan
yuu.nakajima@is.sci.toho-u.ac.jp

Reiko HISHIYAMA
School of Creative Science and Engineering
Waseda University
Tokyo, Japan
reiko@waseda.jp

Takao NAKAGUCHI
Department of Social Informatics
Kyoto University
Kyoto, Japan
nakaguchi@i.kyoto-u.ac.jp

I. Introduction

The internationalization of economic and social activities is forcing people who use different languages and belong to different cultures to collaborate far more often. While English is regarded as the international language, many people are unable to participate adequately via English communication. An analysis of the status of multilingual communications among such people is required. We apply gaming simulation methods to analyze this topic. This paper defines gaming simulation (or simply gaming) as an experiment performed in virtual space, wherein avatars controlled by humans and agents controlled by computer programs play assigned roles.

Gaming environments have been developed mainly as educational tools and experimental tools for research. Some studies have attempted to develop a gaming environment for analyzing communication. Unfortunately, costs are excessive because the conventional approach is to build a different gaming environment for each problem. For decreasing development cost, some researchers have created gaming platforms that can be reused for similar experiments. However, such gaming environments remain clearly specialized for particular domains.

In this study, we design an environment that can game multilingual communication online by describing just simple game scripts. This advance is significant as people with domain knowledge of the applied problem are not always computer experts.

II. Architecture of gaming environment

The architecture of the proposed gaming environment is depicted in Fig. 1. Its design separates the development of an online game system from game design, and so enables experts of different domains to cooperate in efficiently develop a gaming system and conducting game sessions.

Web UI system manages the interaction between game players, facilitators, and agent players as output by *Game definition interpreter*. The game definition interpreter loads and interprets game definitions to conduct a game simulation. *Game Database* stores all information of the game and systems such as user information, game definitions, and game results.

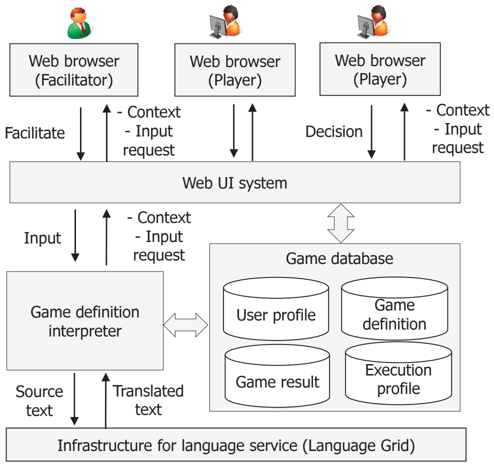

Figure 1. Architecture of gaming environment

The environment is home to human players and agent players. Each human player is controlled by a human subject while agent players are controlled by a computer program. Each player is assigned a scenario and plays their own role. The gaming system displays the forms for inputting decisions in accordance with the game definition. Human players are required to input their decisions. Agent players automatically follow the decisions calculated for them.

The game definition interpreter has a library for calling "Language Grid"[1] which is a global language service. Language Grid is an infrastructure for language services on the Web. The library helps to implement multilingual communication games.

III. Multilingual communication gaming of YMC-Viet Project

One example of multilingual communication is the YMC (Youth Mediated Communication)-Viet Project managed by Kyoto University and the NPO PANGAEA[2]. In the YMC-Viet Project, Vietnamese youth and Japanese agricultural experts share knowledge about rice crops over the Internet (Fig. 2).

The Vietnamese Youth selects questions and the Japanese Expert in rice answers them. The answers are translated

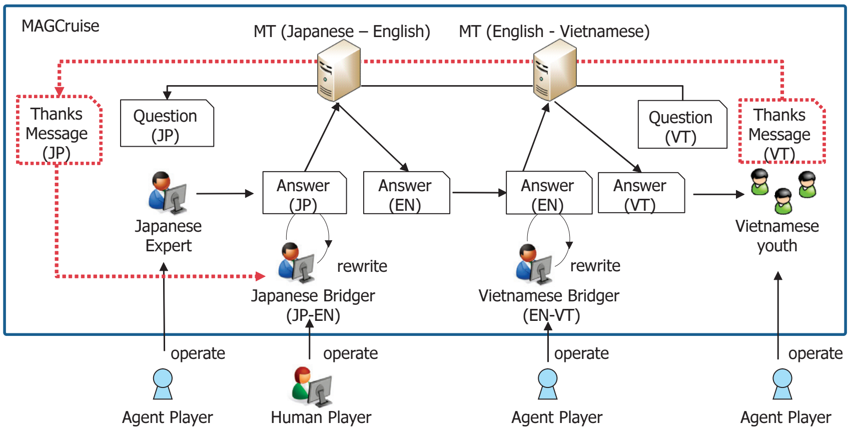

Figure 2. Multilingual communication gaming of YMC-Viet Project

by Japanese-English machine translation (MT) service and English-Vietnamese MT service and sent to the Vietnamese Youth. The Bridgers are volunteers and rewrite the sentences to raise translation quality or repair poorly translated messages as needed.

We use gaming to evaluate incentive strategies for Bridgers so as to motivate their assistance. We added thanks message from Vietnamese Youth to the Japanese Bridger (JP-EN) (red dashed arrows in Fig. 2). We tested the effects of the thanks messages, which express thanks from service recipients to service suppliers as valuable feedback, by gaming this system.

In this game, human players are assigned the role of the Japanese Bridger. They acted as the Bridger and receive feedback from the Vietnamese Youth as simulated by agent players. The Bridger is asked to edit original Japanese sentences to make them simple to translate. He can receive back-translated (JP→EN→JP translated) sentences for reference.

The execution flow of a game is as follows. At first, the game is initialized and the gaming system waits for a participant. When a player logs into the system, the first round of the game is started. After starting the game, all players play their roles as specified by the game definition. The game is stopped when the final round is completed.

A part of the game scenario is as follows.

```
(define (def:game-scenario)
  (def:player ’JP-Expert ’agent)
  (def:player ’JP-Bridger ’human)
  (def:player ’VT-Bridger ’agent)
  (def:player ’VT-Youth ’agent)

  (def:rounds 10
    (def:stage ’answer-question
      (def:task ’JP-Expert ’answer)
      (def:task ’JP-Bridger ’bridge-jp-en)
      (def:task ’VT-Bridger ’bridge-en-vt))
    (def:stage ’feedback
      (def:task ’VT-Youth ’decide-feedback))))
```

The game scenario describes a part of workflow of the multilingual communications. *def:game-scenario* consists of definitions of players and workflows. *def:player* indicates the player name and player type. *def:stage* includes serialized tasks. *def:task* includes actor and action which corresponds with a function call. *def:rounds* bundles stages and repeats itself at specified intervals.

A part of the function *bridge-jp-en*, which express the action of the Japanese Bridger, is as follows.

```
(define (bridger-jp-en ctx self)
  (ui:request-to-input self:name
    (ui:form
      (<div>
        (<h3> "Original sentence")
        (<blockquote> ctx:orig-text)
        (<h3> "Back translated sentence")
        (<blockquote>
          (langrid:BackTranslation-backTranslate
           "GoogleTranslate" "ja" "en" ctx:orig-text)))
      (ui:text
        "Edit orig. sentence to make it
                           simple to translate."
        ’revised ""))
    (lambda (revised)
      (set! self:revisedSentence revised))))
```

ui:request-to-input sends HTML form to the human player and processes the response. *langrid:BackTranslation-backTranslate* function calls the back translation service on Language Grid.

After the experiment, the players answered a questionnaire. We found that more than 80 percent of the human subjects welcomed the feedback from service recipients, regardless of whether the messages indicated the success or failure of the knowledge transfer. Additionally, we assessed the correlation of the self-assessed performance level and the expectation of feedback messages. As a result, we found that service suppliers expect to be thanked if they feel that they had performed well.

References

[1] T. Ishida, *The language grid: Service-oriented collective intelligence for language resource interoperability.* Springer Science & Business Media, 2011.

[2] K. Kita, T. Takasaki, D. Lin, Y. Nakajima, and T. Ishida, “Case study on analyzing multi-language knowledge communication,” in *2012 International Conference on Culture and Computing (Culture and Computing 2012)*, 2012.