The Problem
Program Syntax is too difficult for beginners.
Program in different languages is a headache even for experienced programmers.

The System
- Every time a command in natural language is received by PiE, it will be decomposed word by word into discrete text chunks, denoted w1, w2, ... wN.
- Syntax analysis will be conducted with the assistance of the PiE script. Each of these text chunks will be matched up with a predefined token.
- Programs in the LOGO programming language will be synthesized based on the syntax.
- Incomplete programs will be parsed by rule sequence, and meanwhile, the parameters in these commands will be substitute with the tokens from the input sentences.

Eliza, a primitive prototype of natural language processing, plays the role of a therapist to communicate with patients. The input sentences are processed with a predefined script, where there are two basic types of rules: the decomposition rules and the reassemble rules. When a sentence is typed in, it will be decomposed into pieces according to the decomposition rules and then based on one of the reassemble rules, answer in natural language will be generated automatically.

LOGO programming language is a graphic-oriented education programming language (Feurzeig and Papert 1967). The well-known application of LOGO language is the Turtle Graphics (Papert 1971), by means of which, children are capable of exploring the programming world based on their own logic world.

We have developed a system called Programming in English (PiE) to synthesize domain-specific programming languages from conversations between users and computer. We adopted Eliza for a novel application on program synthesis. Our preliminary experience shows that PiE can assist and enhance programming experience of novices as well as experienced programmers.

Evaluation
We collected 100 natural language descriptions of the PiE commands from these volunteers to test the PiE script that we design. Figure 1 displays the PiE command descriptions and the success ratio to synthesize them from the natural expressions. The system achieves average success ratio of 88.4% in most of the commands that supported by the LOGO language.

To evaluate the accessibility of our system, we set Figure 2 as the target for every user that participated in our test and they are requested to using both the LOGO language and natural language to draw such a graph after 3 minutes reading the tutorial. PiE saves 43.8% and 22.4% of time, respectively, for novices and experienced programmers to draw the same graph.

Conclusions
- We propose a novel way for domain-specific program synthesis based on Eliza and recall the importance of natural language programming with gradually mature techniques in NLP and easy access to programming devices for end-users.
- We have realized program synthesis in the LOGO programming language from English conversations between users and computer. Programs can be synthesized with very few constraints on the input natural language command.
- We have achieved a first step in natural language programming for education use. Program logic can be learned by end-users with no experience during the interaction with PiE system regardless of using complicated programming language like C++ or Java.

Xiao Liu, Dinghao Wu
College of Information and Sciences Technology