

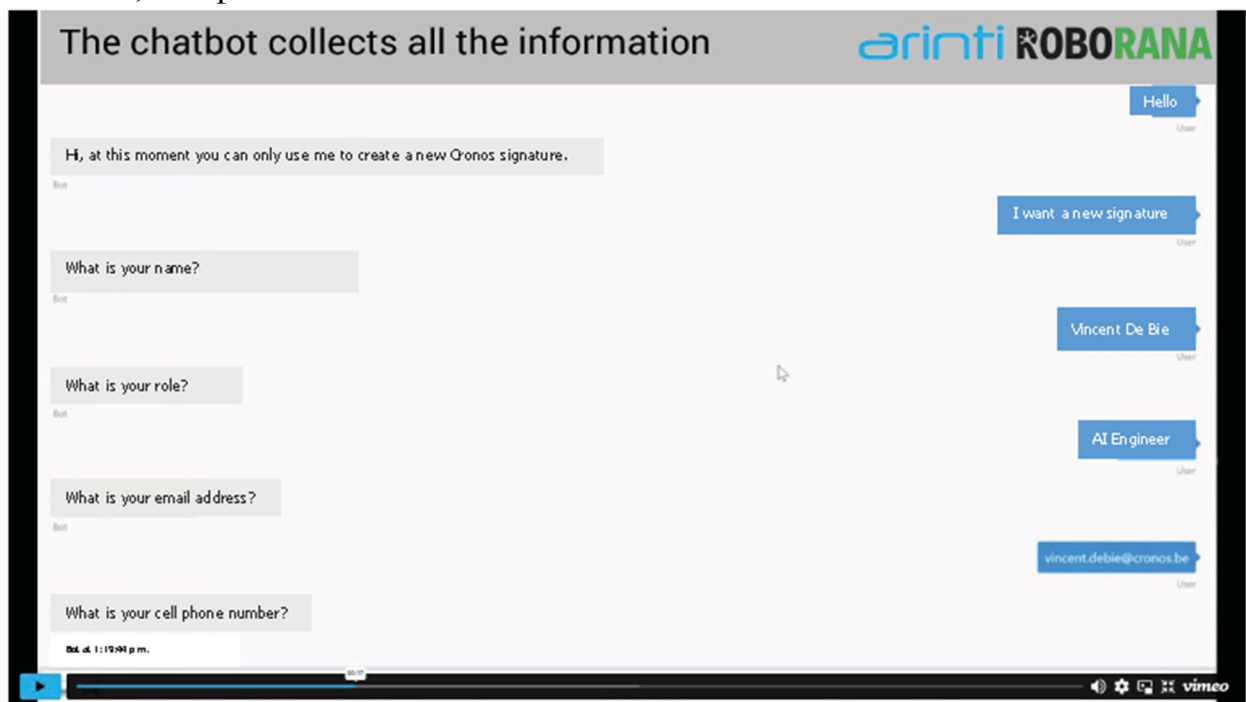
Overview of My “Simple Chatbot” Code and Video

I have developed a Chatbot with the equivalent functionality of Symons reference as described by Symons Blog¹ and demonstrated by the Symons Video² in support of my Declarations submitted as Ex2001 in IPR Nos. 2023-01368, 2023-01379, and 2024-00065. The following is a summary of what capability is demonstrated in the Symons cited material, and how my chatbot replicates that functionality.

Chatbot capability demonstrated in Symons

As indicated in the Symons video, the specific steps performed by the Symons chatbot are:

1. Chatbot waits for an initial message from the chatbot user
2. Chatbot announces that it can only “create a new Chronos signature”
3. Chatbot waits for another message from the chatbot user
4. Chatbot then prompts the user for four pieces of information: name, role, email, and phone number.



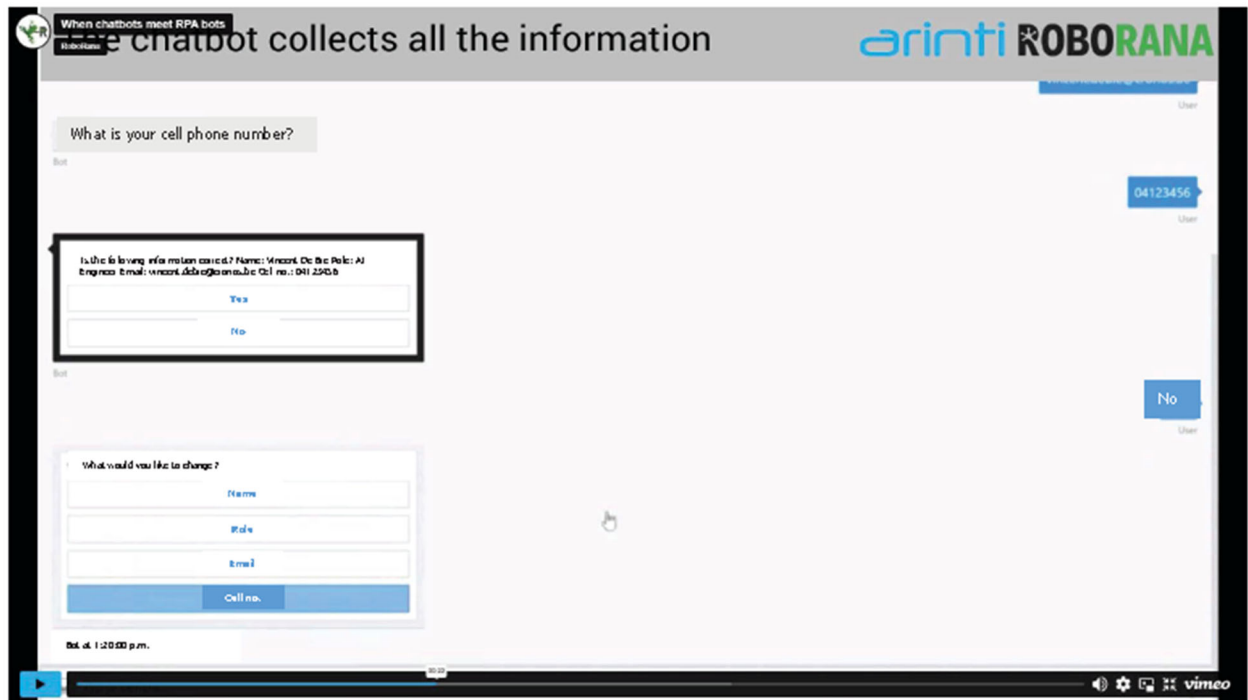
¹ Ex1004, S. Symons, When Chatbots meet RPA bots, <https://medium.com/roborana/when-chatbots-meet-rpa-bots8b22579f49a>, Medium (Oct. 19, 2017)

² Screenshots of S. Symons, RPA meets chatbot: Automatically generate e-mail signature, <https://vimeo.com/238927879>, Vimeo (Oct. 19, 2017)

Overview of My “Simple Chatbot” Code and Video

After the chatbot prompts the user for those four data fields it proceeds with:

5. Chatbot asks if the user wishes to change the information.
6. User may change one or more of the values previously enter and/or approves then approves that information.
7. Once the user has approved the information they have entered, that information (name, role, and phone number) is transmitted to the RPA server.



(Exhibit 1006 Page 9)

Dr. Beigi has asserted that a chatbot inherently uses Natural Language Processing (NLP) since the Chatbot presents the user with English language text and the user can respond with English language text. (Ex2003, 139:3-140:10). However, this is not the criteria for Natural Language Processing. Natural Language Processing would require that the chatbot comprehend, or at least recognize, some portion of the messages sent to and from the user.

As of 2017 there were many types of chatbots that existed, many of which did not have Natural Language Processing such as simple rules based chatbots, or even simpler data entry chatbots which prompted user for information. In the case of the simple rules based chatbot, there would be very specific set of conditions in which the chatbot could request information and/or understand the messages input by the user. In the case of a simple data entry chatbot, it is not necessary for the chatbot to understand any information or requests sent by the user.

Overview of My “Simple Chatbot” Code and Video

If the function being performed by the chatbot was very limited, the chatbot would not require a more complex chatbot that processes and recognizes English language requests or responses. For example, if all that was required was for the user to input four data fields (e.g. name, role, email, phone number) implementing a complex chatbot with Natural Language Processing would be a significant overkill both in capability and in cost.

In addition, as described in the Symons blog, the only capability being demonstrated in the RoboRana / Cronos Group showcase is the ability to integrate chatbots with AI technology such as a Robotics Process Automation bot (or RPA bot). There is no mention of either the type of chatbot used or that it has the capability for Natural Language processing. And, neither the blog nor the video demonstrates any chatbot capability other than to enter and edit those data fields.

For example, when the user enters the message “I want a new signature”, there is no indication that the chatbot even understood that request. It could easily be programmed to proceed to the prompt “what is your name” regardless of what the user typed.

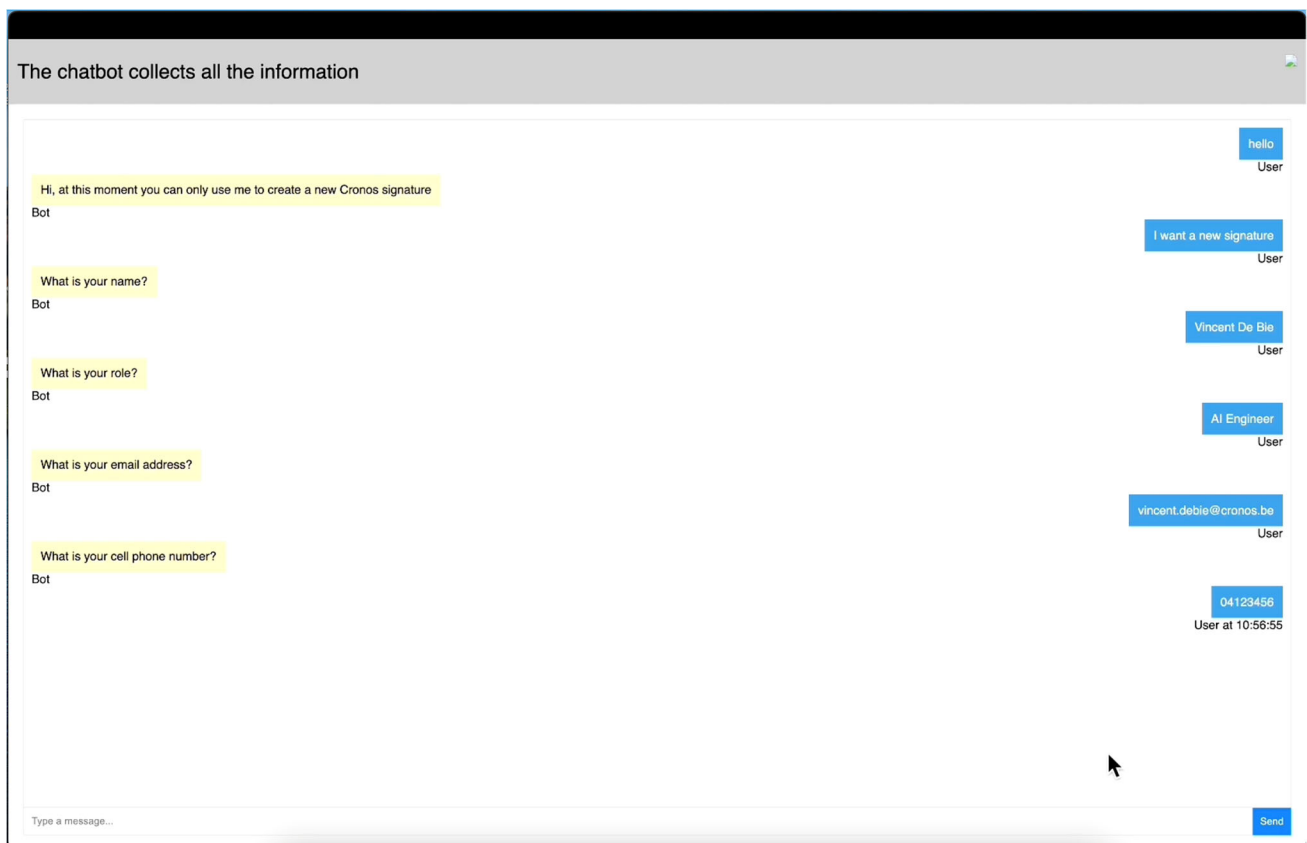
The only time that the video indicates that the chatbot actually recognizes or understands anything input by the user is in the dialog boxes asking the user to verify or change the information they previously entered. As Dr. Beigi indicated (Ex2003, 199:11-200:19), dialog boxes with simple button choices do not demonstrate the capability of Natural Language Processing.

Overview of My “Simple Chatbot” Code and Video

Chatbot capability of my chatbot program (inputting same user input as Symons)

As indicated in the “Simple Chatbot” video (Ex2008), the specific steps performed by my chatbot are:

1. Chatbot waits for an initial message from the chatbot user
2. Chatbot announces that it can only “create a new Chronos signature”
3. Chatbot waits for another message from the chatbot user
4. Chatbot then prompts the user for four pieces of information: name, role, email, and phone number.



Overview of My “Simple Chatbot” Code and Video

As shown in the HTML below, the chatbot simply proceeds through a series of prompts, one after another. In response to each of the last four prompts (i.e. “What is your name? ... What is your cell phone number”) the chatbot simply collects whatever the user types in (e.g. immediately after prompting for the name (return ‘What is your name?’) the HTML simply collects whatever message the user types in the userName variable. (userName = message;).

```
function getBotResponse(message) {
  switch (state) {
    case 'initial':
      state = 'new_signature';
      return 'Hi, at this moment you can only use me to
create a new Cronos signature';
    case 'new_signature':
      state = 'get_name';
      return 'What is your name?';
    case 'get_name':
      userName = message;
      state = 'get_role';
      return 'What is your role?';
    case 'get_role':
      userRole = message;
      state = 'get_email';
      return 'What is your email address?';
    case 'get_email':
      userEmail = message;
      state = 'get_phone';
      return 'What is your cell phone number?';
    case 'get_phone':
      userPhone = message;
      state = 'change_question';
```

Overview of My “Simple Chatbot” Code and Video

After my chatbot prompts the user for those four data fields it proceeds with:

1. Chatbot asks if the user wishes to change the information.
2. User may change one or more of the values previously enter and/or approves then approves that information.
3. Once the user has approved the information they have entered, that information (name, role, email, and phone number) is transmitted to the RPA server.

The screenshot displays a chatbot conversation within a web browser. The chatbot, identified as 'vincent.debie@cronos.be' with phone number '04123456', initiates the process by asking, 'Do you want to change anything? (yes/no)'. The user responds 'Yes'. The chatbot then presents a menu titled 'What would you like to change?' with options: 'Change Name', 'Change Role', 'Change Email', and 'Change Phone'. The user selects 'Change Phone'. The chatbot prompts, 'Please enter the new phone number:', and the user enters '0412345678'. The chatbot confirms, 'Phone number updated to 0412345678. Do you want to change anything else? (yes/no)', and the user responds 'No'. Finally, the chatbot sends a message: 'Sending your information to Server: Name: Vincent De Bie Role: AI Engineer Phone: 0412345678 Email: vincent.debie@cronos.be'. The user responds 'Thank you! Your signature is being created. Final Information: Name: Vincent De Bie Role: AI Engineer Phone: 0412345678 Email: vincent.debie@cronos.be'. The chatbot concludes with 'Bot at 10:57:14'. The interface includes a 'Type a message...' input field and a 'Send' button.

Note, that the behavior of my chatbot is identical to the chatbot in the Symons reference. The information that is collected by the chatbot is sent to a Server (e.g. an RPA Server). This is done the most common HTML method known as POST a URL.

The following shows the function `someWebsiteFunction` that sends those four parameters (name, role, phoneNumber, Email) in a URL to be “POSTed” to the “`RPAServer.com`” server as highlighted below.

```
function someWebsiteFunction(UserName, userRole, userPhone, userEmail) {
```

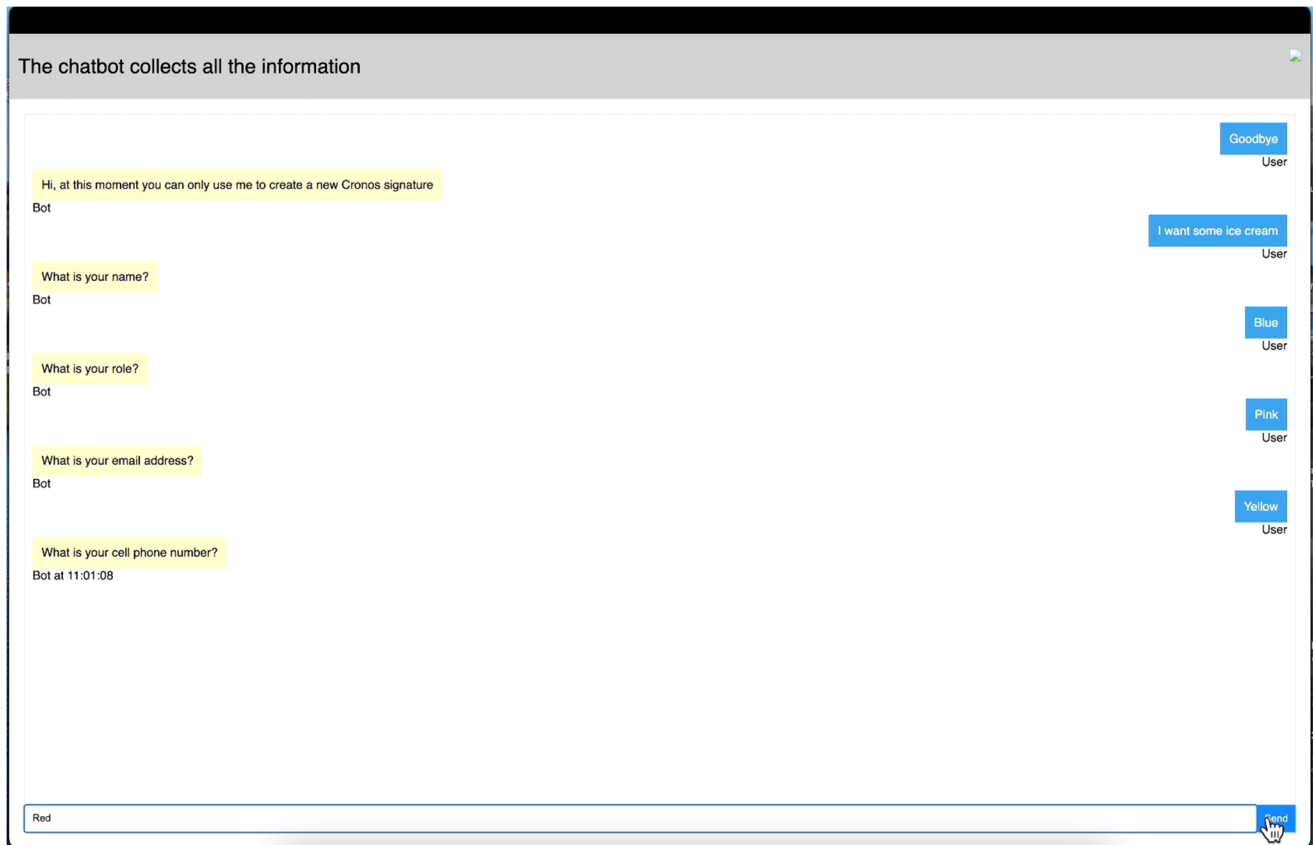
Overview of My “Simple Chatbot” Code and Video

```
const url =  
'https://RPAServer.com/api?Name=$userName}&Role=${userRole}  
&Phone=${userPhone}&Email=${userEmail}';  
const data = { name, role, phone: phoneNumber,  
Email };  
  
fetch(url, {  
  method: 'POST',  
  headers: { 'Content-Type':  
'application/json' },  
  body: JSON.stringify(data)  
})  
  .then(response => response.json())  
  .then(responseData => {  
    console.log('Success:', responseData);  
  })  
  .catch(error => {  
    console.error('Error:', error);  
  });  
}
```

Chatbot capability of my browser program (inputting different user input as Symons)

By running my chatbot again, this time with different user input, we see that the chatbot still prompts the user for the same information regardless of what the user types. For example, when notified that user can only create a Cronos signature, the user requests ice cream. The chatbot has no understanding of this response, but it is not necessary since it continues with the same steps as before, which is to collect information to be included in the signature.

Overview of My “Simple Chatbot” Code and Video



This is typical behavior of simple data entry chatbots, and even rules based chatbots, that do not have Natural Language Processing. In this case, Natural Language Processing is not required since the Symons chatbot only performs one task, to collect four pieces of information (whatever the user chooses to provide) and send them to the Server to be included in a new signature.

Overview of My “Simple Chatbot” Code and Video

As indicated below, this is exactly what my chatbot does. After an example of changing the phone number from “Yellow” to “Purple”, it sends those four pieces of information to the Server to be included in the signature.

The screenshot displays a chatbot interface with a black header bar. The chat area has a white background with messages from the bot and user. Bot messages are in white boxes with black text, and user messages are in yellow boxes with black text. The interface includes a 'Send' button at the bottom right and a 'Type a message...' input field at the bottom left.

Bot: Thank you! You have entered this information: Name: Blue Role: Pink Email: Yellow Phone: Red Do you want to change anything? (yes/no)

User: Yes

Bot: What would you like to change?

User: Change Name

User: Change Role

User: Change Email

User: Change Phone

Bot: Please enter the new phone number:

User: Purple

Bot: Phone number updated to Purple. Do you want to change anything else? (yes/no)

User: No

Bot: Sending your information to Server: Name: Blue Role: Pink Phone: Purple Email: Yellow

Bot: Thank you! Your signature is being created. Final Information: Name: Blue Role: Pink Phone: Purple Email: Yellow

Bot at 11:01:28

Type a message... Send