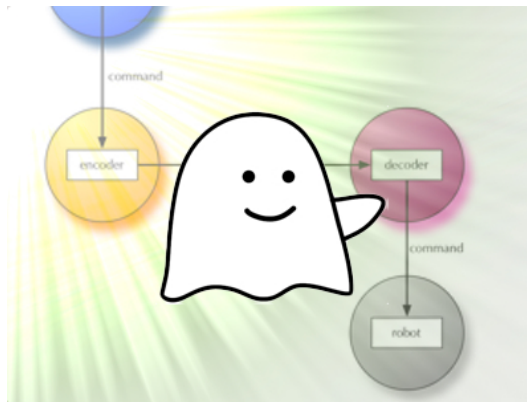


スペーストーキー問題

© 2014 結城浩

<http://www.hyuki.com/codeiq/>

2014 年 4 月



1 概要

あなたはスペーストーキー社にプログラマとして雇われました。スペーストーキー社は普段、遠い星で動作している探査ロボットにコマンドを送って調査活動を行っています。ところが、中継基地が壊れてしまい、普通にコマンドを送ることができなくなりました！

途中の通信では変換アルゴリズムが不明の文字列処理が行われています。その変換アルゴリズムを推測して、探査ロボットに送れるコマンドを見つけることがあなたのミッションなのです！

2 会話

あなた「探査ロボットにコマンドを送る仕事と聞きましたが、いったいどういうことでしょうか……」

依頼者「順を追って説明しましょう。わがスペーストーキー社では、ここから遠く離れたある星に探査ロボット (robot) を設置して調査活動を行っています」

あなた「はい」

依頼者「わが社の研究センター (research center) は、英小文字列のコマンド (command) を探査ロボットに送り込んで、探査ロボットを操作します。コマンドを送るために、途中の星に中継基地 1 (encoder) と、中継基地 2 (decoder) を設置してあります。通信構成図を示しましょう (図 1)」

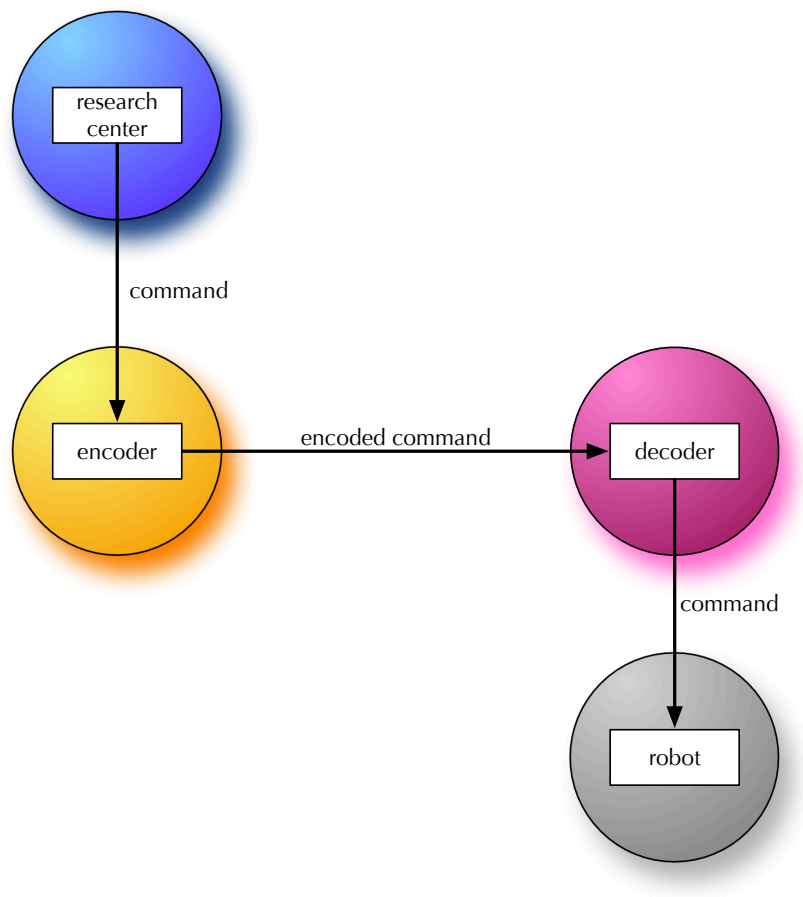


図1 通信構成図（通常）

あなた「なるほど、わかります」

依頼者「研究センターから中継基地 1 (encoder) へコマンドを送ると、中継基地 1 (encoder) は、そのコマンドを **ある形式のデータ** (encoded command) に変換して中継基地 2 (decoder) へ送ります」

あなた「ある形式のデータ、とは？」

依頼者「実は、encoded command の形式をわれわれは知りません」

あなた「は？」

依頼者「歴史的な事情によります。ともかくわれわれは encoded command の形式を知らない。しかし、これまでは、それで特に問題はありませんでした。なぜなら、中継基地 2 (decoder) が、encoded command をもとのコマンドに復元して探査ロボットに送ってくれていたからです」

あなた「これまでは……とは？」

依頼者「不幸なことに、中継基地 2 (decoder) が故障してしまったのです (図 2)。そのため、こちらから探査ロボットへ正しいコマンドは送れなくなってしまいました。中継基地 2 (decoder) が故障して、中継基地 1 (encoder) の出力が探査ロボットに直接送られてしまうようになったからです！」

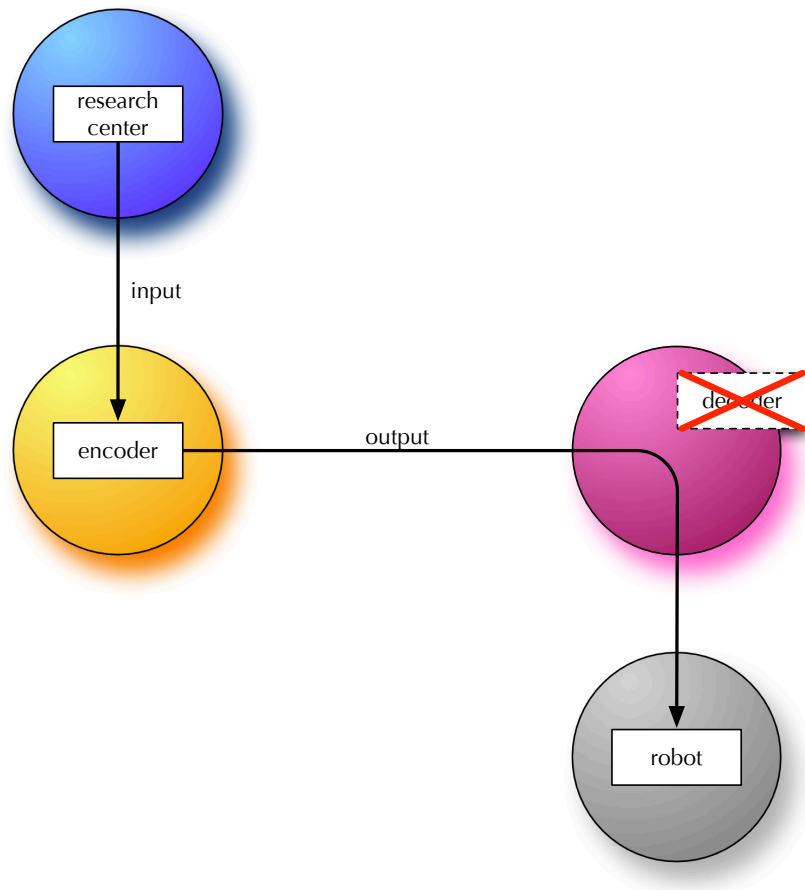


図 2 通信構成図 (故障時)

あなた「なるほど。それでは、中継基地 1 (encoder) の方も、素通しにすればいいですね！」

依頼者「それができていればあなたを呼ぶことはなかったでしょう (キリッ)」

あなた「そうですか。では中継基地 1 (encoder) のソースコードを調べさせてください。そうすれば encoded command の形式を把握することができるかもしれませんから」

依頼者「申し上げにくいのですが、ソースコードもドキュメントも存在しません」

あなた「えっ！……では、私に何をどうせよと」

依頼者「幸い、中継基地 1 (encoder) には **エコーバックモード** があります。これを使えば、あなたが送った入力 (input) がどのように中継基地 1 (encoder) で変換されるかは正確にわかります」

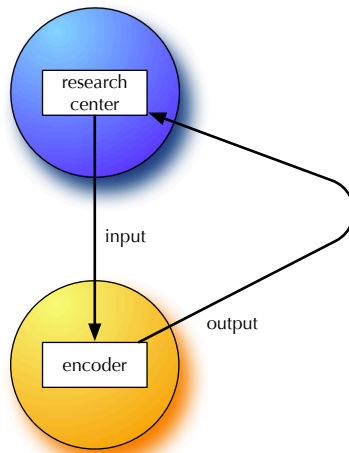


図3 通信構成図（エコーバックモード）

あなた「ちょっと待ってください。それではということですか。ソースコードもドキュメントもないから、中継基地 1 (encoder) がどうい変換アルゴリズムで動いているかはわからない。エコーバックモードを使って変換アルゴリズムを推測せよと」

依頼者「端的にいえばそういうことです。変換アルゴリズムを推測して、output が正しいコマンドになるような input を考えてほしいのです。それがわかれば、探査ロボットを操作することができるからです」

あなた「(中継基地 2 を先に修理しろよ……)」

依頼者「何かおっしゃいましたか。大丈夫ですか。では、さらに具体的にお話ししましょう。あなたには、エコーバックモードになった encoder にアクセスするための **Web API** をお渡しします」

あなた「Web API?」

依頼者「そうです。以下の URL にアクセスしてください」

`http://spacetalky.textfile.org/api.cgi?input=INPUT`

あなた「これは?」

依頼者「この URL で *INPUT* の部分には encoder への input となる文字列を与えます。*INPUT* に与えられるのは **英小文字列** のみで、その長さは 1 以上 1024 以下です」

あなた「なるほど」

依頼者「この URL にアクセスすれば、HTTP のレスポンスとして encoder からの output が返されてきます。Content-type は text/plain です。output として返されるのも **英小文字列** です。ただしエラーの場合には?という一文字が返されます」

あなた「例を挙げていただけますか」

依頼者「たとえば input として foo という文字列を与える場合には、次の URL にアクセスします。すると、output は faob になります」

```
http://spacetalky.textfile.org/api.cgi?input=foo
↓
faob
```

依頼者「何らかのエラーが起きた場合には、一文字?だけが返されます。たとえば以下では、input として Foo を与えましたが、大文字が混じっていたのでエラーとなりました」

```
http://spacetalky.textfile.org/api.cgi?input=Foo
↓
?
```

あなた「なるほど。理解しました」

依頼者「あなたにやっていただきたいのは、**output** が正しいコマンドになるような **input** を、コマンドごとに見つけ出すという仕事です」

あなた「なるほど。たとえば、**faob** が正しいコマンドだとすれば、それを作り出す input として **foo** を見つけ出すという仕事になりますね」

依頼者「その通りです。ただ、**faob** は正しいコマンドではありませんけれど。探査ロボットの正しい **コマンド一覧表** (**words.txt**) は別途ファイルとしてお渡しします」

あなた「わかりました。あっ！ ちょっと待ってください。output が正しいコマンドになるような input が必ず存在する保証はあるのでしょうか」

依頼者「そのような保証はありません。何しろわれわれは変換アルゴリズムを知らないのですから。正しいコマンドを作り出す input がない場合には、その旨を明記していただければよろしいです。逆に、正しいコマンドを作り出す input が複数個あった場合でも、一つだけ見つければけっこうです」

あなた「わかりました。がんばってみましょう」

ミッションはこのようにして始まりました……。