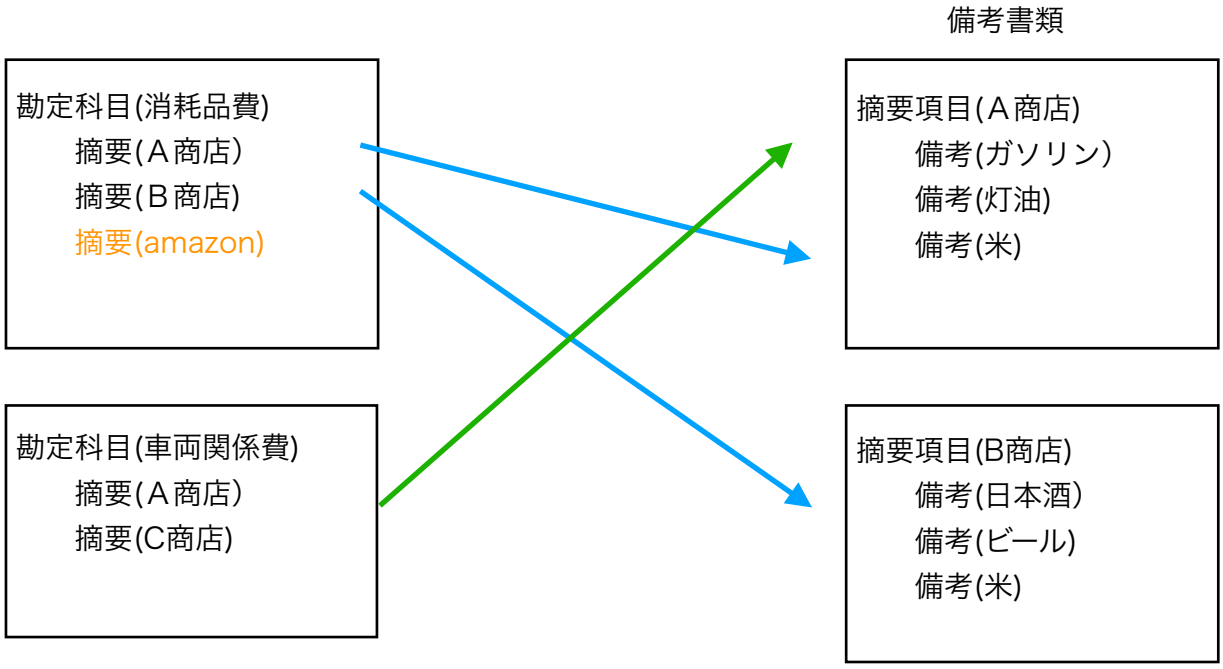

この様な状態を考える。
ここへ消耗品費―摘要-amazonで備考-米を入れようとする時どうするか
備考書類では、摘要項目のかぶりは無いようにしたい

摘要　amazon　を消耗品費で探す。

ある

備考書類で摘要　amazonの　子の備考を求めて表示する
終わり

ない

摘要　amazon　を備考書類で探す。

ある

備考書類で摘要　amazonの　子の備考を求める

ある

備考を表示する　終わり

ない

備考書類に摘要項目　amazonを作る
その子に、備考ノードを追加
追加したノードに属性-bikouを追加
bikouに米を入れる
終わり

ない

備考書類のamazonに備考ノードを追加する
追加したノードに属性-bikouを追加
bikouに米を入れる
終わり

XMLにおける検索について考えてみる

put revXMLMatchingNode(tgBikouTree, "/",, "tekiyou", attributeValue, -1) into uKotae --備考書類から摘要を検索する

revXMLMatchingNode この関数を使うと完全一致で検索されるようです

次のようなスクリプトで検証してみます。

```
on mouseUp pButtonNumber

  --xmlデータのIDを記録してあるフィールドから読み込む
  put line 1 of field "ID" into gDenTree

  ---変数に検索したい言葉を入れて検索する
  ask "完全一致で探します。探す言葉を入れてください"
  put it into stext
  put textEncode(it,"utf8")into stext--2バイト文字はエンコードしておく


  put revXMLMatchingNode(gDenTree,"/",,"bikou",stext,-1)into tAttNode--検索語で属性bikouの中を検索した結果

  put tAttNode&return after tNodeList --結果を変数に収納(ノード)

  put revXMLNodeContents(gDenTree, tAttNode) into tAttCont--見つけたノードの内容を得る

  put tAttCont & return after tNodeList--内容(金額)

  put revXMLAttribute(gDenTree, tAttNode, "bikou")&return after tNodeList--属性の内容(備考名)


  put textDecode(tNodeList,"utf8")into field “data4"

end mouseUp
```

ここでサーチテキストとして「パーツ」をいれてみると返り値は
/仕分/摘要[189]/備考[1]
3300
パーツ
つまり、189番目の摘要の一番目の備考で、値は3300、内容はパーツ　と言うこと。これは一番最初に見つかったアドレスです。

次以降に「パーツ」があるかどうか調べるにはどうするか。

下のスクリプトでは、サーチテキストを孫の中から探して、もしあればそのひとつ上の子と、そのノードを表示させます。

```
on mouseUp pButtonNumber

  --ノードを一つずつ調べてリストアップした中にサーチテキストがあるかどうかを調べる
  --みつけたら、そのノードの一つ下の属性をすべてリストアップする。これを繰り返す

  --xmlデータのIDを記録してあるフィールドから読み込む
  put line 1 of field "ID" into gDenTree

  ---変数に検索したいことばを入れて検索する
  ask "探す備考を入れてください"," "
  put it into sarchtext--エンコードしてはいけない

  put textEncode("仕分","utf8")into Shiwake
  put textEncode("摘要","utf8")into Teki
  put textEncode("備考","utf8")into Biko


  put revXMLNumberofChildren(gDenTree,"/",Teki,-1) into tNum--仕分けノードを数える
  put empty into fld data2
  put empty into field data3
  put empty into field data4

  repeat with x=1 to tNum--ここは摘要のクラス

    put Shiwake&"/"&Teki&"["&x&"]" into tNode--数えられた分だけ摘要をリストアップしていく
    put revXMLAttribute(gDenTree, tNode, "tekiyou")into tekiName--この名前を得る
    --摘要を順番に調べてその備考をリストする
    put revXMLAttributeValues(gDenTree, tNode,,"bikou", return, -1)into kotae
    if kotae is empty then next repeat--孫が無ければ次のループへ
    put textDecode(kotae,"utf8")into uKotae--孫をデコードしてサーチテキストを探す
    if uKotae contains sarchtext then
      put kotae &return after tkotaeList--あれば保存へ
      put tNode&return after tNodeList
      put tekiName&return after tekiList
    end if
  end repeat

  put textDecode(tNodeList,"utf8")into tNodeList
  put tNodeList into fld data2
  put textDecode(tekiList,"utf8")into tekiList
  put tekiList into fld data3
```

```
  put textDecode(tkotaeList,"utf8")into tkotaeList
  put tkotaeList into fld data4

end mouseUp
```

revXMLAttribute(gDenTree, tNode, "tekiyou")

revXMLAttributeValues(gDenTree, tNode,,"bikou", return, -1)

この2つはAttributeが子の属性(この場合摘要の名前)を得ます。
Valuesがついた方はtNodeの孫の属性(この場合備考の名前)を全てリストアップ
valuesでは、return(or、tab)をつけないとエラーになります。ー1をつけることで、子以下を全てリストアップ出来ます。

下はサーチテキストとして「パーツ」を入れて、属性をリストアップしたもの。
左がノード、真ん中が摘要の内容、右が孫のリスト