

化学プラントの初期設計におけるコンピュータ支援

Intelligent Systems for the Conceptual Design of Chemical Plant

○伊東 正敬*, 山下善之*, 鈴木 睦*

○Masahiro Itou*, Yoshiyuki Yamashita*, Mutsumi Suzuki*

*東北大学

*Tohoku University

キーワード： コンピュータ支援 (Computer-Aided Design), 概念設計 (Conceptual Design),
協調設計 (Concurrent Design), マルチエージェント (Multi-Agent)

連絡先： 〒 980-77 仙台市青葉区荒巻字青葉東北大学 大学院 工学研究科 化学工学専攻 鈴木研究室
伊東 正敬, Tel.: (022)217-7266, Fax.: (022)217-7293, E-mail: mito@pse.che.tohoku.ac.jp

1. はじめに

一般に化学プラントの設計は、「概略設計」と「詳細設計」から構成される。「概略設計」は要求仕様に対して実現する手段の機能と構造の概略を明らかにする段階（概念設計）と、要求仕様を満たす実現手段を定量的に評価し、実現手段の基本的な詳細を決める段階（基本設計）とに分けられる。「詳細設計」は基本設計を基に、設計対象の製造が可能なレベルまで詳細を決める段階である。

最近ではこれらの設計においても、さまざまなプロセスシミュレータや最適化ツールが活発に利用されている。特に基本設計の段階では定常プロセスシミュレータなどがよく用いられている。しかし、定常プロセスシミュレータを使用する前の段階、即ち概念設計の段階においては、コンピュータ支援が十分になされているとはいえないのが現状である。その理由として、

- 概略設計は熟練した設計者の知識に大きく依存する
- 対象が漠然としている
- 化学プラントは内部情報が複雑である

などが挙げられる。一方、化学プラントの概略設計で扱われる知識は化学工学的な一般的知識であることが多いため、知識ベースとしてそれをうまく管理できればコンピュータ支援ができる可能性がある。しかし、要求を考えて、それを機能ブロック化する段階までは、人間の仕事である。そこで、要求を機能ブロックで表現したところから、基本設計で用いられる定常プロセスシミュレータにフローシートとして入力できる段階までの設計を支援するようなコンピュータ支援に注目し、その支援システムの構築を目的とした検討を行った。

2. 基本概念

最近のソフトウェア開発においては協調プログラミングの研究が様々な分野で行われている。これは、複数のプログラム（エージェント）が協調して問題を解決するような形でプログラムを記述するスタイルのことである。複数のプログラムが協調することによって、一つのプログラムだけでは解きにくい問題が解ける場合がある。

化学プロセスの設計においては、環境規制の範囲内であるか、コストの最適化、安全性や操作性についてはどうか、といった多くの設計目的が存在する。そこで、本研究ではエージェントの概念を用いて、複数の対等な設計目的を複数のエージェントに振り分けることによって、各エージェントが協調して競合を解消しながら実現可能な解を導く方法を採用した。

ここで、問題になるのは各エージェントをどのような手法で協調させるかである。これに関しては、多数の計算主体が集まって目標を達成するマルチエージェントシステムが提案されている。

マルチエージェントシステムでは、エージェントが協調してシステム全体の目的を達成することから、各エージェントがその動作について他のエージェントの動作とどのように干渉するのか、そういった干渉をどのように回避するのか、といった競合を解消する方法が重要な問題になる。その場合、

- エージェント間で通信を行なう際の手順（プロトコル）を決めておき、それによって情報をやりとりし競合状態を解消する。各々の行動計画を何らかの基準によって階層構造に分類し、それに沿って通信を行なうことにより、お互いの競合状態を認識し修正案を作成することでお互いの動作を協調させる方法。¹⁾

- システム全体の構成を工夫することにより、競合状態を解消するための仕組みをシステムで持っているもの。

各エージェントが自由に読み書きできる「黒板」にシステム全体で共有する情報を集めておき、各エージェントがそれを読み、書き換えることによりエージェント間で直接情報のやりとりをせずに競合の解消を可能にする「黒板」システム。²⁾

等が提案されている。本研究では、黒板システムを採用した。

次にプロセスの機能についてであるが、機能とはプロセスの期待される入出力のことであり、物理的構造はこの際問われないものである。ここで、主機能を構成するサブ機能、サブ機能を構成するサブサブ機能といった具合に機能に着目してプロセスを階層的に分解した方がプロセスの初期設計の段階では役に立つ。化学工学の場合、「反応」、「分離」といった機能が主機能の例として挙げられる。機能ブロックの一例を図1に示す。

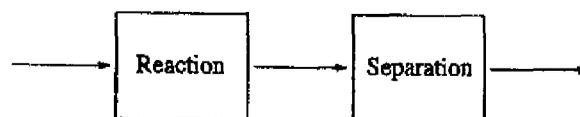


Fig. 1 機能ブロック.

機能の分解は必要に応じて意味のあるレベルまで分解する。それが装置レベルなのか、装置を構成する部品レベルなのかを一概に判断することは難しい。本研究では装置レベルまで分解した。図2に「分離」についての機能分解ツリーの例を示す。

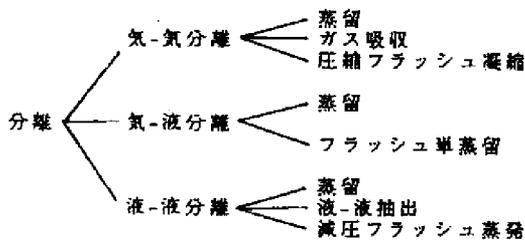


Fig. 2 機能分解ツリー.

3. システム構成

本研究で提案するシステムの構成を図3に示す。

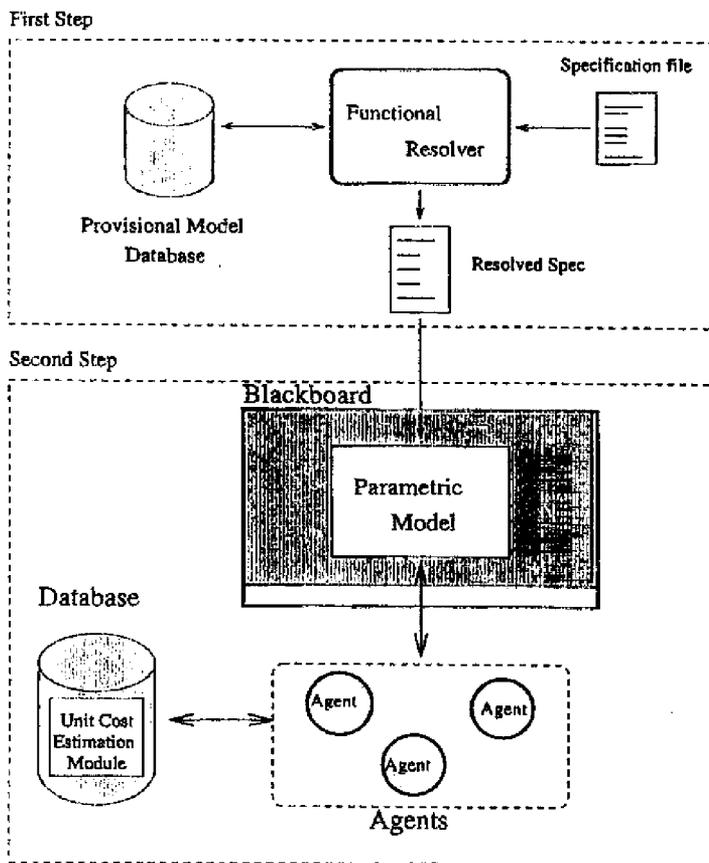


Fig. 3 システム概略図.

3.1 実装方法

図3に示した本システムの処理の手順は、まずステップ1で要求仕様ファイルから「Resolved Spec」ファイルを対話的に半自動生成させる。ステップ2で「Resolved Spec」の情報をもとに、要求を満たす実現可能な解を複数のエージェントにより協調的に求めていく。なお、黑板とエージェント間の情報交換は、ソケットによるプロセス間通信により実現させた。シミュレーションは、UNIXワークステーション (Pentium/133MHz) 上で行ない、プログラミング言語はCを用いた。次に、システムの各部分について説明する。

3.2 要求仕様ファイル

要求仕様ファイルのテンプレートのインスタンスを図4に示す。

```

Component_in: (BENZENE,n-HEPTANE,SULFUR_DIOXIDE)
Composition_in: (0.6,0.4,0.000001)
Component_out: (BENZENE,n-HEPTANE)
Composition_out: (0.98,0.98)
Flowrate(kg-mol/hr): 400
#Flowrate(kg/hr):
Feed_pressure(atm): 1

Feed_Phase: Liquid(bubble_point)
Liquid_mole_fraction: 1

Function: Separation

#Utility Condition
Coolant: Water
Inlet(°C): 30
Outlet(°C): 40
Heat_Source: Steam
Heat_Source_Temperature(°C): 400

```

Fig. 4 要求仕様ファイル.

要求仕様には、入出力情報、要求する機能、ユーティリティー条件をフレーム表現で記述する。尚、Feed_Phaseの項に記述できるフレーム値は、

- Liquid(bubble_point) : 沸点の液
- Gas(dew_point) : 露点の蒸気
- Mixed : 気液混合状態

の3つとする。

3.3 Provisional Model Database

このデータベースには、簡易モデルを記述しておく。次の2つを簡易モデルと定義する。

- 機能のデータベース
要求された機能とその機能を達成させるために使用可能な装置群の関係をフレーム表現で記述したもの。内容は図2に対応させる。
- 初期必要条件のデータベース
装置のコスト推算の際にユーザーによる入力が最低限必要な初期操作条件名を各装置ごとにフレーム表現で記述したもの

簡易モデルのデータベースの内容を図5、図6に示す。

```
Function: Separation

  Available_Device(Liquid):
    Distillation,L-L_Extraction,Flash

  Available_Device(Gas):
    Absorption,Distillation,Flash

  Available_Device(Mixed):
    Distillation,Flash
```

Fig. 5 機能のデータベース。

```
Distillation:
  operating_pressure

L-L_Extraction:
  operating_pressure,operating_temperature,
  solute_component,solvent,solvent_temperature,
  distribution_coefficient

Absorption:
  operating_pressure,abs_oil,
  oil_temperature,oil_rate

Flash:
  operating_pressure
```

Fig. 6 初期必要条件のデータベース。

3.4 Functional Resolver

機能分解プログラム (Functional Resolver, 以下FR) は、要求仕様ファイルと簡易モデルのデータベースから、「Resolved Spec」ファイルを半自動生成させるプログラムである。ここで、「Resolved Spec」ファイルはシステムへの入出力情報と、要求された機能を達成させるために使用可能な装置群の情報を記述したものである。「Resolved Spec」ファイルの生成手順について、具体例を挙げて説明する。FR はまず、図4の要求仕様ファイルの Function の項から主機能を認識し、さらに Feed_Phase の項からフィードの相状態を認識する。次に以上の情報から if~then ルールにより、主機能に対するサブ機能を推論し、図5のデータベースから機能を達成させるのに使用可能な装置群を抽出する。次に、使用可能な装置群について、図6のデータベースから得られる初期操作条件をユーザーに入力するよう促す。この処理によって得られる「Resolved Spec」ファイルの例を図7に示す。

```
Component_in: (BENZENE,n-HEPTANE,SULFUR_DIOXIDE)
Composition_in: (0.500000,0.400000,0.000001)
Component_out: (BENZENE,n-HEPTANE)
Composition_out: (0.980000,0.980000)
Flowrate(kg-mol/hr): 400.000000
Feed_pressure(atm): 1.000000
q_value: 1.000000

#Utility condition
Coolant: Water
Inlet(°C): 30.000000
Outlet(°C): 40.000000
Heat_Source: Steam
Heat_Source_Temperature(°C): 400.000000

#Available Device
Available_Device: (Distillation,L-L_Extraction,Flash)

#Operating Condition
operating_pressure[atm](Distillation): 1.000000
operating_pressure[atm](L-L_Extraction): 8.000000
operating_temperature[°C](L-L_Extraction): 125.000000
solute_component(L-L_Extraction): BENZENE
solvent(L-L_Extraction): DEC
solvent_temperature[°C](L-L_Extraction): 125.000000
distribution_coefficient(L-L_Extraction): 0.431000
operating_pressure[atm](Flash): 0.500000
```

注) 液-液抽出において
BENZENE-HEPTANE-DEC (ジエチレングリコール) 系に
おける BENZENE の分配係数の値は参考文献 (3) から引用した。
分配係数の単位は質量分率単位である。

Fig. 7 Resolved Spec.

3.5 Cost Estimation Database

このデータベースには、以下のものを用意した。

- 物性データベース
各種成分の物性データをテキストファイル形式で用意
- 物性推算モジュール
密度（気体、液体）、粘度（気体、液体）、蒸発潜熱、エンタルピー（気体、液体）を推算するモジュールを用意
- 機器コスト推算モジュール
蒸留塔、液-液抽出塔、吸収塔、断熱フラッシュ分離器の機器コストを推算するモジュールを用意

3.6 Parametric model

本研究におけるパラメトリックモデルとは、「Resolved Spec」の情報と各装置の内部情報を含むデータ構造である。

3.7 黒板

黒板は各エージェントに対して、パラメトリックモデルをやりとりし、それを保持する存在である。情報がそこに集められているので、各エージェントが、黒板を読み書きするだけでエージェントの間で情報交換を行うことが可能であり、エージェント同士が直接情報交換を行うようなシステムよりも情報のやりとりが容易に実現できる。図8に、黒板の状態遷移の様子を示す。

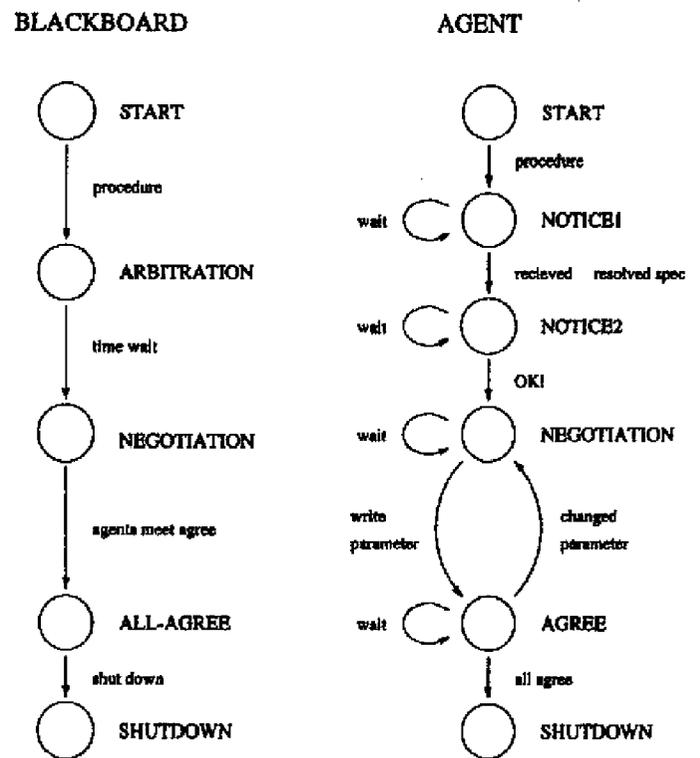


Fig. 8 黒板とエージェントの状態遷移の様子。

各状態では次のような処理を行う。

- START
「Resolved Spec」ファイルを読み込み、エージェントとプロセス間通信を行うための手続きを行う。
- ARBITRATION
各エージェントから参加する旨のメッセージを受け取り、参加するエージェントに「Resolved Spec」ファイルの内容を伝える。一定時間経過後、次の状態に移る。
- NEGOTIATION
エージェントの要求に従って、現在黒板に書かれているパラメトリックモデルの内容を知らせたり、書き換えたりする。この時、エージェントがパラメトリックモデルを書き換えなかった場合、黒板は、そのエージェントが

現在書かれているパラメトリックモデルに賛成した、と解釈する。また、その内容が書き換えられた場合、前の案に賛成していたエージェントにそのことを知らせる。そして、全エージェントが賛成するまで、この一連の処理を繰り返し行なう。全エージェントが賛成すると合意が成立した、と解釈し、次の状態に移る。

- ALL-AGREE

全エージェントが賛成すると、そのことを各エージェントに知らせる。

- SHUTDOWN

各エージェントにシステムを停止することを伝え、プログラムを終了する。

以上のように今回行なったシミュレーションでは、黒板がシステム全体の同期をとることも行なっている。

3.8 エージェント

エージェントは複数の if~then ルールで記述された推論機構を持っており、if 部は、黒板から読み取ったパラメトリックモデルの値、then 部はその値に対してどういう処理をすべきかという規則が記述されている。エージェントの状態遷移の様子を図 8 に示す。各状態では以下のような処理を行なう。

- START

黒板とプロセス間通信を行うための手続きを行なう。

- NOTICE1

参加する意思表示のためのメッセージを送り、黒板から、「Resolved Spec」ファイルの内容を読み出す。

- NOTICE2

黒板から、OK の指示を受け取るまで待つ。

- NEGOTIATION

黒板から読み出したパラメトリックモデルの値を評価し、自分の設計目的を達成できるような処理を行なう。処理を行なう際、必要に応じて装置のデータベースから必要なモジュールを逐次呼び出して処理を行なう。
1、ここで、他のエージェントの提案と自分の提案が異なっている場合には、各エージェントの優先度（4 節、参照）に基づいて妥協するのか、妥協しないのかを決定する。

— 妥協する場合には、妥協案を生成する

— 妥協しない場合には、自分の案を黒板に送る

2、他のエージェントの提案と自分の提案が同じ場合には OK のメッセージを黒板に返す。

- AGREE

黒板に先ほどのパラメトリックモデルに他のエージェントが賛成したかどうかを問い合わせ、他のエージェントがそれを書き換えたなら、一つ前の状態に戻る。全員が賛成したら、次の状態へ移る。

- SHUTDOWN

黒板から SHUTDOWN のメッセージを受け取るとプログラムを終了する。

実際にシステムを実現する場合においては、情報交換を行なう際に様々な障害が生じることが予想されるが、本研究では、エージェントの数はシステムの起動から終了まで変化せず、それぞれが扱う情報は黒板からの読み出し、書き込みの際には、変化しないという仮定をおいた。

4. ケーススタディ

化学プラントにおいて要求される機能は「分離」、「反応」など様々である。しかし、プラント全体を考えるのは、非常に困難であるので、今回は、「分離」という機能に注目してケーススタディを行った。尚、成分数は2成分系において適用する。また、「分離」機能を達成させるのに使用可能な装置は、3-5節において用意した装置を対象とする。

エージェントの目的が、以下に場合についてシミュレーションを行なった。

- 1) エージェント1：機器コストを最小にしようとするエージェント
- 2) エージェント2：装置出口の環境規制成分の濃度を規制値以下に抑制するエージェント
- 3) エージェント3：装置の一般的な物理的制約条件をみるエージェント

各エージェントの処理について述べる。

エージェント1が機器コストの最小化を行なう際に、変えるパラメーターは次の通りである。

- 蒸留塔：(実際還流比/最小還流比)の比率
- 液-液抽出塔：抽出液量
- 吸収塔：吸収液量
- フラッシュ分離器：槽の操作圧力

エージェント2は、装置出口の環境規制成分の濃度(体積基準)を評価して、規制値を上回っていれば、規制値に押さえるような装置をつけて、そのコストを求める。今回は、規制成分に亜硫酸ガス(SO_2)を設定した。また、亜硫酸ガスの除去装置として水による吸収塔(充填塔)を用意した。規制値は、 $0.5[\text{ppm/hr}]^4$ とした。

エージェント3は、熱交換器(コンデンサー、リボイラーも含む)の伝熱面積について、次に示す

一般的な制約条件を違反していたら、違反を解消するような処理を行なう。

- 伝熱面積： $1000[\text{m}^2]^5$

伝熱面積が $1000[\text{m}^2]$ を越えていたら、熱交換器に入る処理量を半分にして機器コスト計算を行ない、熱交換器を2つにする

ここで、エージェントの競合の解消は次のように行なった。

- 1) あらかじめ各エージェントに優先順位をつけておく
- 2) 競合した場合、優先順位の低い方が妥協案を生成する。妥協案生成のためのルールはあらかじめ記述しておく

優先順位は、エージェント2 > エージェント3 > エージェント1と設定した。次に妥協案生成のためのルールを示す。本例題の場合、エージェントの数が3つなので競合する組み合わせは次の3通りである。

- 1) エージェント1とエージェント2が競合した場合：
エージェント1はエージェント2の提案に賛成する。
- 2) エージェント1とエージェント3が競合した場合：
エージェント1は、先の自分の提案に対して、熱交換器の伝熱面積が制約以下になるように(実際還流比/最小還流比)還流比の比率を下げ、機器コストを計算する(妥協案)。
還流比の比率を下げるのは、コンデンサー、リボイラーの大きさを小さくするためである。その後、相手の提案と自分の妥協案を自分の目的のもとで比較する。自分の妥協案の機器コストの方が安ければ黒板の値を書き換える。しかし、相手の提案の機器コストの方が安ければ相手の提案に賛成する。

- 3) エージェント2とエージェント3が競合した場合
 エージェント3はエージェント2の提案に賛成する。

5. 結果

5.1 本システムによる結果

要求仕様として、図4を用いてシミュレーションを行なった時の結果を次に示す。図4からFRを経て、図7の「Resolved Spec」が生成される。次に、黒板と各エージェント間のメッセージのやり取りを図9に示す。

```

ID 3 Send RS == ARB == Send RS
ID 3 HOW? == ARB == WAIT
ID 1 Send RS == ARB == Send RS
ID 2 Send RS == ARB == Send RS
ID 3 HOW? == ARB == WAIT
ID 1 HOW? == ARB == WAIT
ID 2 HOW? == ARB == OK
ID 3 HOW? == ARB == OK
ID 1 HOW? == ARB == OK
---
ID 2 Send PA == NEGO == WAIT
ID 3 Send PA == NEGO == WAIT
ID 1 Send PA == NEGO == --ATT
ID 2 Send PA == NEGO == --ATT
ID 3 Send PA == NEGO == --ATT
ID 1 HOW? == NEGO == --AGR changed
ID 2 HOW? == NEGO == --AGR changed
ID 3 HOW? == NEGO == --AGR wait
ID 1 Send PA == NEGO == --ATT
ID 2 Send PA == NEGO == --ATT
ID 3 HOW? == NEGO == --AGR changed
ID 1 HOW? == NEGO == --AGR changed
ID 2 HOW? == NEGO == --AGR wait
ID 3 Send PA == NEGO == --ATT
ID 1 Send PA == NEGO == --ATT
ID 2 HOW? == ALL AGREE == OK
ID 3 HOW? == ALL AGREE == OK
ID 1 HOW? == ALL AGREE == OK
check_ID OK
ID 2 HOW? == SHUTDOWN == NOT
ID 3 HOW? == SHUTDOWN == NOT
ID 1 HOW? == SHUTDOWN == send shutdown
  
```

Fig. 9 黒板とエージェント間のメッセージのやり取りの様子。

図9において、「エージェントのID番号とエージェントが黒板に送ったメッセージ」==黒板の遷移状態==「黒板がエージェントに送ったメッセージ」を表す。

次に、各エージェントが提案したパラメトリックモデルを図10に示す。

○凝留					
Agent	R_ratio	A_cond	A_reb	C	Cost
Agent1	1.35	1112	111	8960	26473
Agent2	1.35	1112	111	0.5	28241
Agent3	1.35	556	111	0.5	29459
Agent1	1.16	996	111	0.5	29299
Agent2	1.16	996	111	0.5	29299
Agent3	1.16	996	111	0.5	29299

ここで、
 R_ratioは(実際運流比/最小運流比)の比率
 A_condはコンデンサーの伝熱面積[m²]
 A_rebはリボイラーの伝熱面積[m²]
 CはSO₂ガスの標準状態における濃度[ppm/hr]
 Costは機群コスト[千円]

○液-液抽出						
Agent	Rate	R_ratio	A_cond	A_reb	C	Cost
Agent1	820	1.0	62	268	8960	11974
Agent2	820	1.0	62	268	0.5	13742
Agent3	820	1.0	62	268	0.5	13742
Agent1	820	1.0	62	268	0.5	13742
Agent2	820	1.0	62	268	0.5	13742
Agent3	820	1.0	62	268	0.5	13742

ここで、
 Rateは抽出液流量[kg-mol/hr]
 なお、液-液抽出でのR_ratio、A_cond、A_rebは抽回回収塔における値である

○減圧フラッシュ蒸発			
Agent	P_ope	C	Cost
Agent1	0.5	8960	545

ここで、P_opeは操作圧力[atm]

注) 減圧フラッシュ蒸発の場合は与えられた条件において、要求された製品組成を満たさなかったため処理が打ち切られた。

Fig. 10 パラメトリックモデルの提案の推移。

さらに、最終的な実行可能解を図 11 に示す。

<p>[蒸留塔]</p> <p>操作圧力：1[atm] フィード温度：86.3[℃] 塔頂温度：80.6[℃] 塔底温度：98.0[℃] 塔頂流量：241.7[kg-mol/hr] 塔底流量：158.3[kg-mol/hr] 段数：54[段] 還流比：2.44[-] 塔高：36.2[m] 塔径：2.76[m] コンデンサー伝熱面積：996[m²] リボイラー伝熱面積：100[m²] 機器コスト：27531[千円]</p> <p>[液-液抽出塔]</p> <p>○抽出塔 操作圧力：8[atm] 操作温度：125[℃] 抽出液成分：DEG(ジエチレングリコール) 抽出液流量：819.6[kg-mol/hr] 塔頂流量：163.3[kg-mol/hr] 塔底流量：1056.3[kg-mol/hr] 段数：8[段] 塔高：3.54[m] 塔径：1.73[m] 機器コスト：2014[千円]</p> <p>○抽回取塔(蒸留)</p> <p>入口組成(BENZENE)：0.224 入口組成(DEG)：0.776 入口流量：1056.3[kg-mol/hr] 操作圧力：8[atm] 塔頂温度：206.8[℃] 塔底温度：334.8[℃] 塔頂流量：224.6[kg-mol/hr] 塔底流量：831.8[kg-mol/hr] 段数：16[段] 還流比：0.18[-] 塔高：11.7[m] 塔径：1.06[m] コンデンサー伝熱面積：62[m²] リボイラー伝熱面積：268[m²] 機器コスト：9960[千円]</p> <p>[亜硫酸ガス除去装置(吸収塔)]</p> <p>入口濃度：8960[ppm/hr] 出口濃度：0.5[ppm/hr] 操作圧力：1[atm] 操作温度：20[℃] 吸収液成分：WATER 吸収液流量：0.90[kg-mol/hr] 充填塔高さ：17.4[m] 機器コスト：1768[千円]</p>
--

Fig. 11 実行可能解.

5.2 定常プロセスシミュレータによる検証

1節でも述べたように、実プラントの設計においては定常プロセスシミュレータが広く利用されている。このソフトを使えば、プロセスフローにそって装置などを指定していき、計算に必要な条件や方法を指定してやるだけで収支計算などを行ってくれる。これを用いて、プロセス設計に必要な装置の大きさや収率、エネルギー消費等を計算することができる。そこで、本システムで得られた実現可能解の妥当性を確認するために図7の入出力条件と図11の操作条件を定常プロセスシミュレータの一つであるPRO/II (PROVISION) 6)に入れてシミュレーションを行った。その結果の一部を図12、13に示す。

蒸留塔については、シミュレータの値と本システムにより得られた詳細な最終解とがよく一致していた。一方、液-液抽出に関してはシミュレータの値との誤差が大きかった。これは、機器コストの概略推定を行うにあたり様々な仮定とヒューリスティックスを用いて推算を行っているためである。しかし、定常プロセスシミュレータの計算が収束していることから本システムで得られた最終解は、最適解ではないが実現可能な解の1つであると推定できる。

6. 結論

本研究では化学プラントの初期設計においてエージェントと知識ベースを用いたコンピュータ支援システムのプロトタイプを作成した。単純なプロセスについては機能ブロックを文書化したところから、定常プロセスシミュレータを用いる前の段階までの設計について、実現可能な解を導くことができた。また、新しいエージェントを追加する場合もエージェントに優先順位をつけ、かつ競合解消のルールを追加することにより、競合を解消しながら実現可能な解を求めることができると思われる。しかし、今回用いたケーススタディは非常に単純なプロセスであるので、より汎用的なシステムにするためには、リサイクル流れや複数の装置からなるプロセスについて検討する必要があると思われる。

参考文献

- 1) Edmund H. Durfee and Thomas A. Montgomery, : A Hierarchical Protocol for Coordinating Multi-agent Behaviors, Proceedings, Eighth National Conference on Artificial Intelligence, 1, 86/93 (1990)
- 2) Victor R. Lesser and Daniel D. Corkill, : Functionally Accurate, Cooperative Distributed Systems, IEEE Trans. SMC, Intelligence, SMC-11-1, 81/96 (1981)
- 3) 別冊化学工業：化学装置設計資料II, 236, 化学工業社 (1983)
- 4) 別冊化学工業：公害防止装置の簡易設計・選定図表, 186, 化学工業社 (1973)
- 5) 高岡 成禎：化学プロセスの評価法, 81, 丸善 (1973)
- 6) PRO/II 4.13 with PROVISION 2.03 リファレンスマニュアル：Simulation Sciences Inc. (1996)

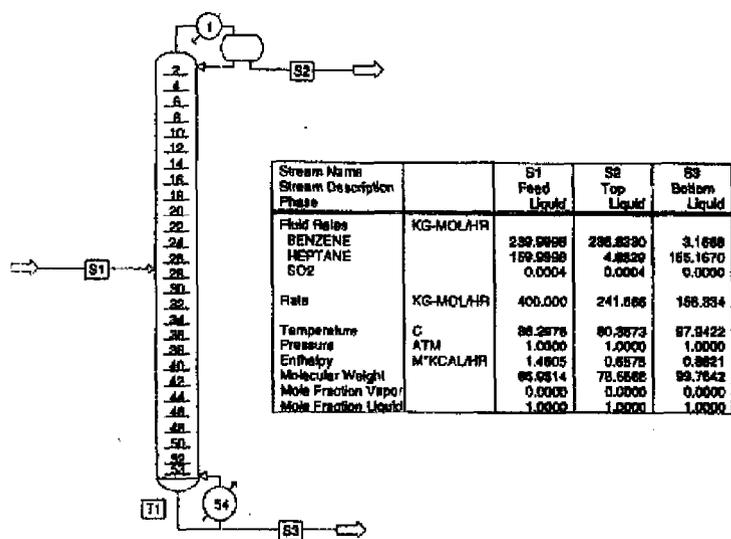


Fig. 12 Report of Distillation.

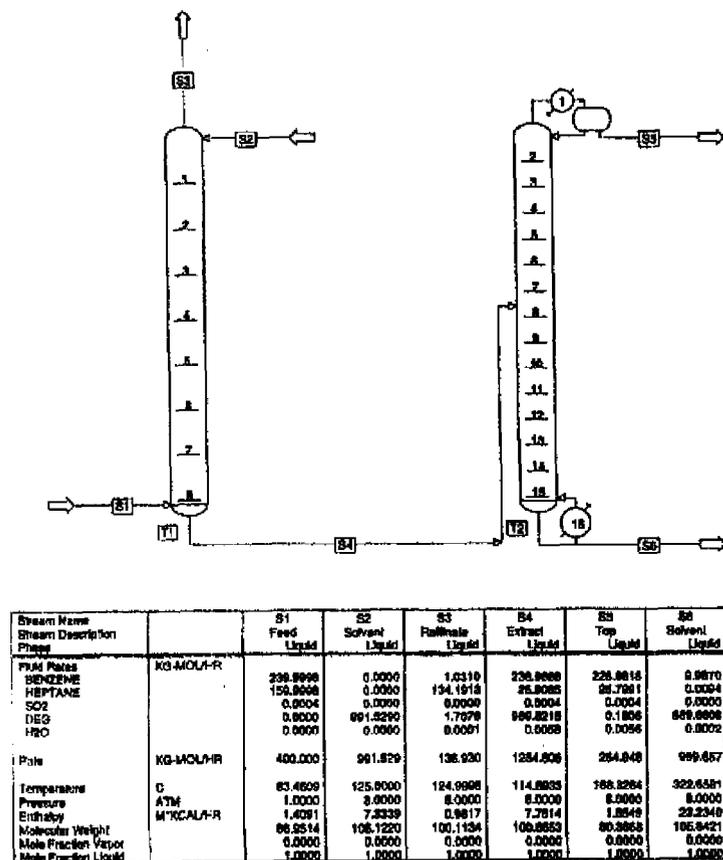


Fig. 13 Report of Extraction.