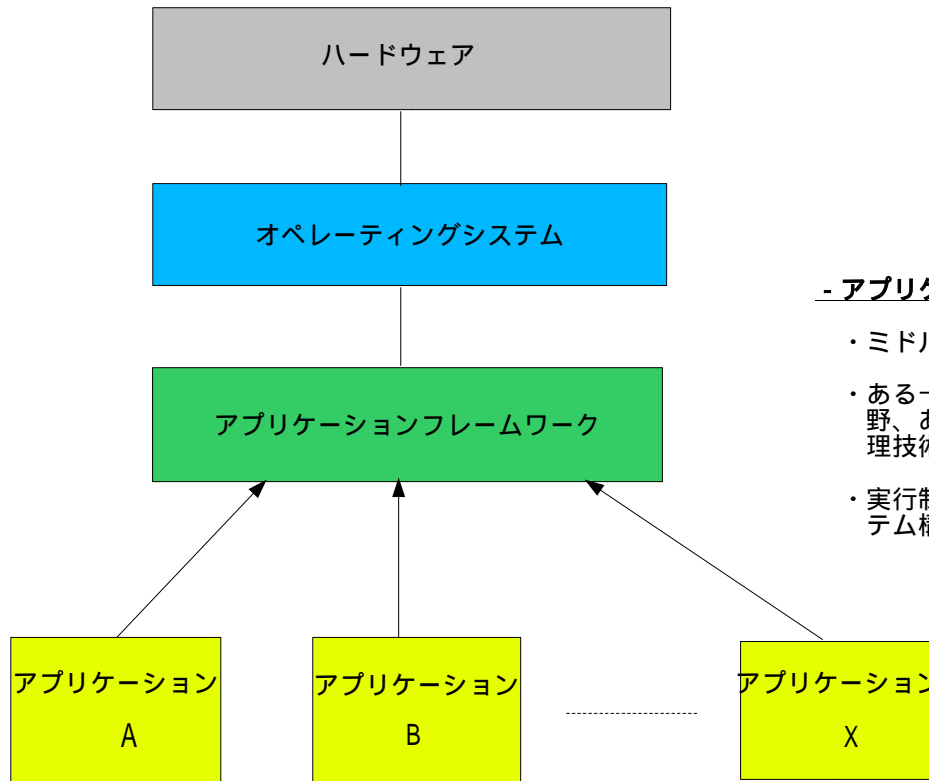


メッセージ駆動型  
アプリケーションフレームワーク  
SWA - AF

(株) ソフトワークスアスカ (SWA)

# アプリケーションフレームワーク



## - アプリケーションフレームワーク

- ・ミドルウェアの一種
- ・ある一定のアプリケーション分野、あるいはある特定の情報処理技術分野を想定
- ・実行制御部を有し、全体のシステム構造を規定する。

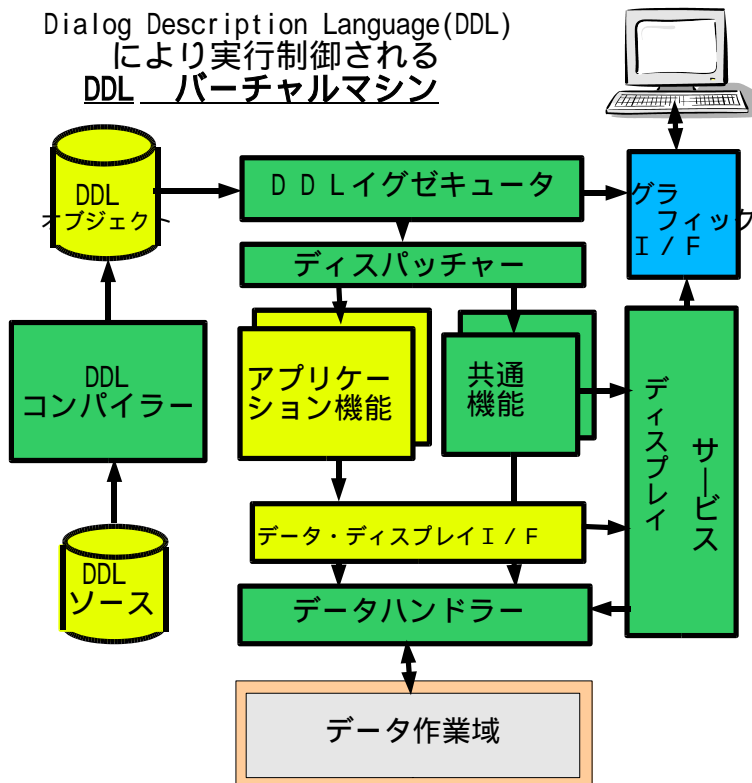
## - アプリケーション部

- ・部品として開発し、フレームワークに組み付ける（フレームワーク型でないミドルウェアはそれ自身が部品としてアプリケーションに組み込まれる）

# アプリケーションフレームワーク事例

## CAD フレームワーク ( 1980 年代 )

Dialog Description Language(DDL)  
により実行制御される  
DDL バーチャルマシン

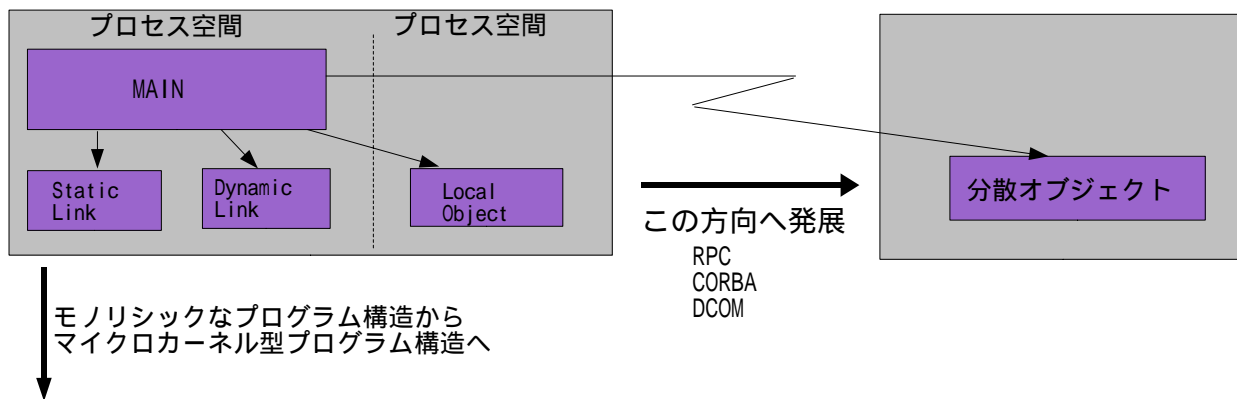


## 開発されたアプリケーション

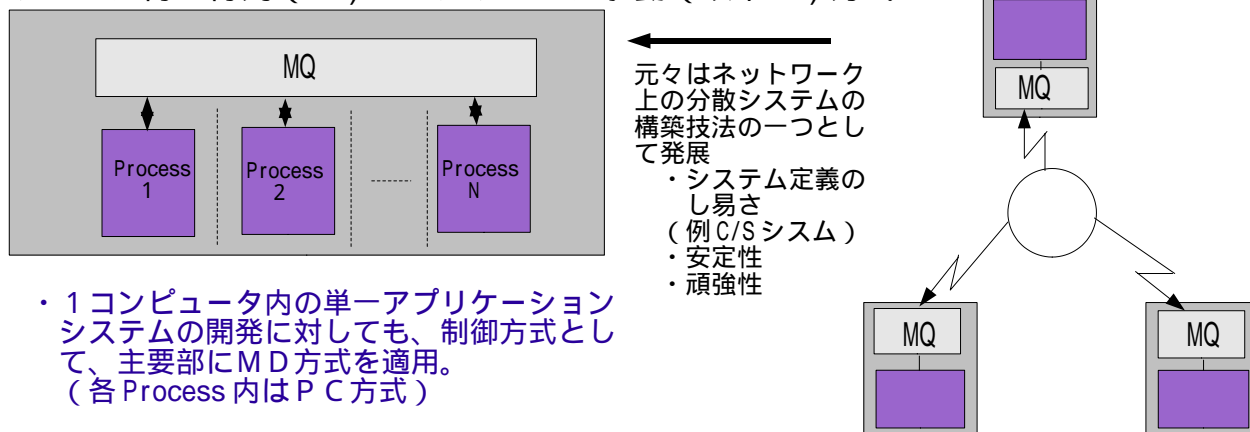
- CAD
  - 汎用機械系 CAD
  - 回路設計 CAD
  - プリント基板 CAD
  - 自動車などの専用 CAD
  - ワイヤーハーネス CAD
- CAM
  - NC
  - 機械加工工程設計
- CAE
  - FEM Pre/Post プロセッサ
- など

# メッセージ待ち行列つきメッセージ駆動方式

プロシージャ呼び出し ( ProcedureCall, 以下 PC) 方式



メッセージ待ち行列 ( MQ) つきメッセージ駆動 ( 以下 MD) 方式



## PC 方式と MD 方式の対比 ( 一般論 / 基本形 )

項目	P C	M D
呼び出し関係	親子関係	双方向性
データ受け渡し	パラメータ (+ クラス変数)	メッセージ
同期 / 非同期	同期型	非同期型
プログラム構造	モノリシック (一枚岩)	マイクロカーネル型 (*)
プロセス	基本は単一プロセス	プリエンティブなマルチプロセスが前提

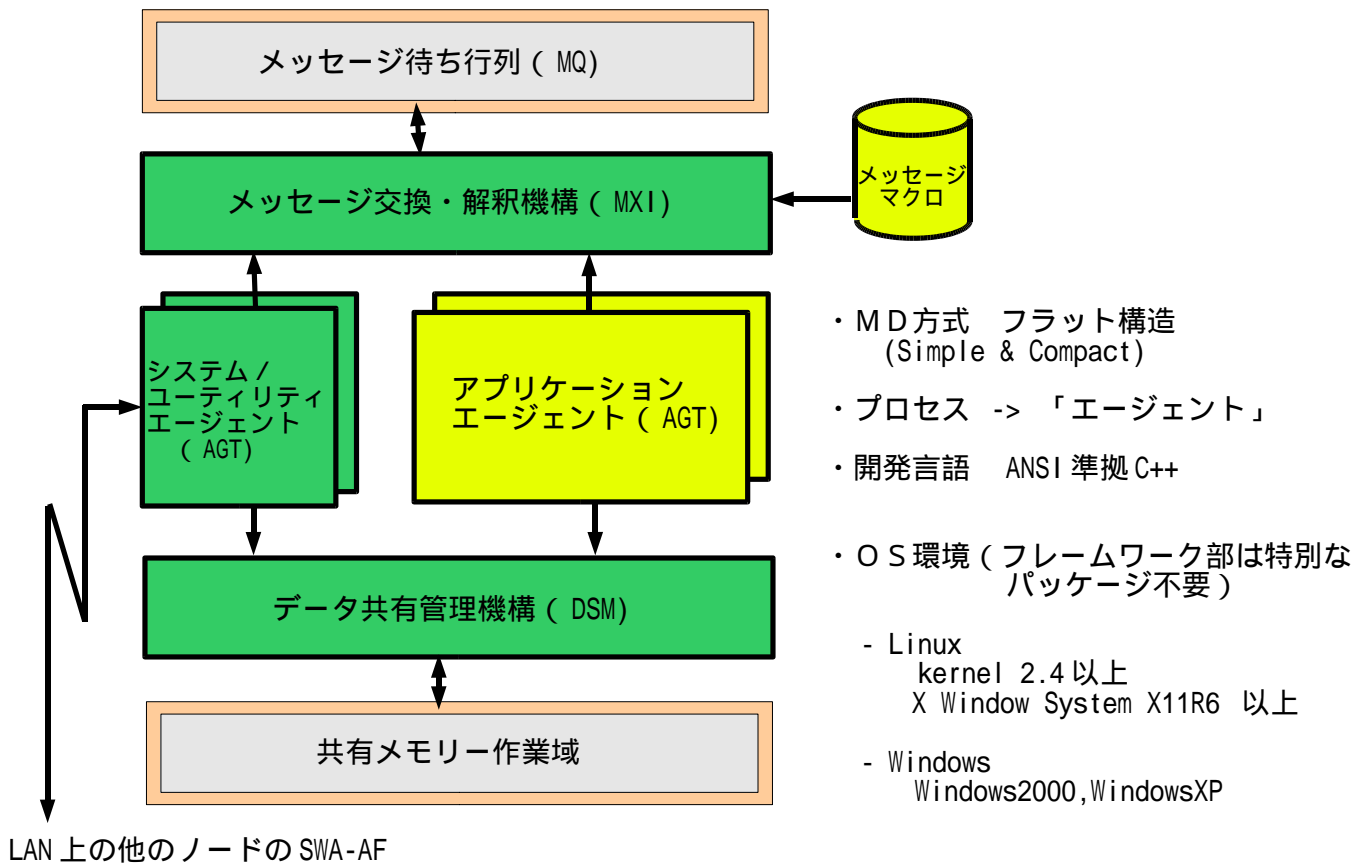
(\*) [ 参考 ]

- <http://yougo.ascii24.com/gh/80/008015.html>
- <http://www.atmarkit.co.jp/icd/root/38/5786938.html>
- <http://yougo.ascii24.com/gh/12/001266.html>



# SWA-AF (SoftWorks Aska Application Framework)

## 基本システム構造



## SWA-AF

### メッセージ基本形

label: message 名 (value) [operand-1, operand-2, ....., operand-n];

ここで

operand-i : operand-i 名 (value) [operand-i1, operand-i2, ....., operand-im]

ここで operand-ik は operand-i と同じ形式 (再帰的表現)

value : 数値、テキストストリング、バイナリストリング

(下線部は省略可能)

#### - メッセージ辞書

個々のメッセージごとに、上記メッセージ基本形に則った「メッセージ辞書」を定義し、処理時に AGT より MXI に引き渡して、メッセージの解釈や、value のバイナリーデータ化などに用いる。(新たなメッセージの追加はそれを受信し、処理する AGT 内で完結する。(そのメッセージを生成・発信する側は別として))

#### - メッセージ例

メッセージ

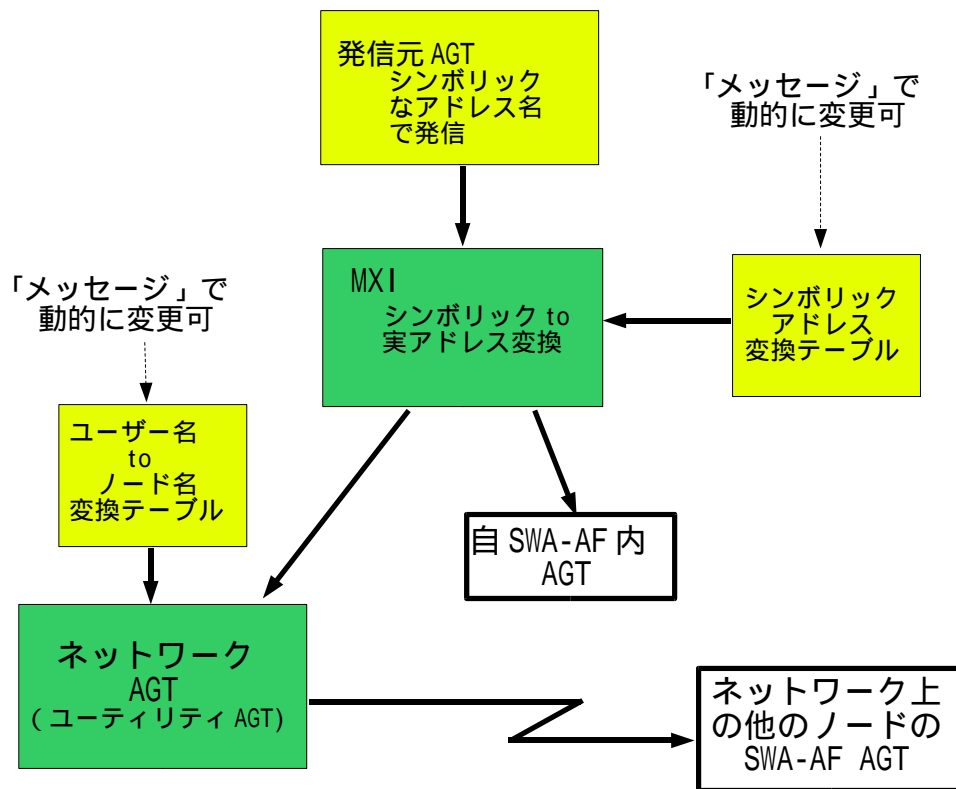
```
pulse_switch(sw01)[origin(90.0,120.0),size(10.0,20.0),label("RESET")];
```

メッセージ辞書

```
static MSGDIC G_pulse_switch[] =  
{  
  {0, "pulse_switch" ,REQ_STR,1,1,NOT_APL, " " ,NAMELENGTH},  
  {1, "origin"      ,REQ_FLT,2,2,NOT_APL , " " ,12},  
  {1, "size"        ,REQ_FLT,2,2,NOT_APL , " " ,12},  
  {1, "label"       ,REQ_STR,1,1,NOT_APL, " " ,12},  
  {-1}  
};
```

## SWA-AF

### メッセージの動的アドレッシング



## SWA-AF

### 開発スタイル (概略)

#### - システム設計

- ・ 必要な AGT とその役割の定義
- ・ 各 AGT ごとに必要なメッセージと、その機能 / 処理内容を定義
- ・ AGT 間のメッセージの送付 / 処理を通じて、机上シミュレーションを行い、システムの挙動を検証する。

#### - プログラム開発

- ・ AGT テンプレートをコピーして各 AGT のボディを作成する。
- ・ メッセージ毎にメッセージ辞書を記述し、メッセージインデックスに登録する。
- ・ AGT ボディ上で、メッセージインデックスに指定したラベルを定義し、そこにメッセージの処理部のコーディングをする。
- ・ 共有メモリーを参照しないメッセージなどでは、ユーティリティ AGT からメッセージをキー入力したり、メッセージマクロファイルを利用して、各メッセージの単体テストがこの段階で可能となる。

# SWA-AF

## アプリケーションイネーブラー

SWA-AF + アプリケーションイネープリング機能

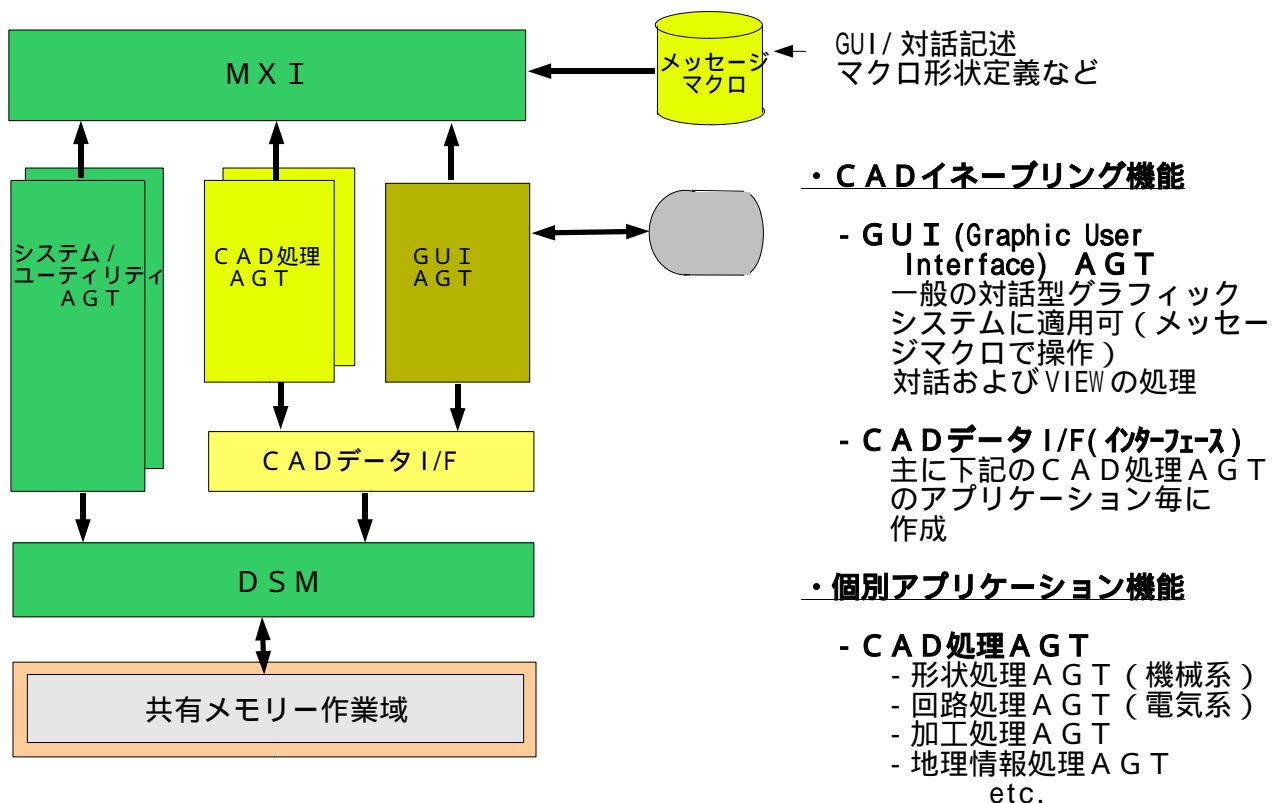
### アプリケーションイネープリング機能

特定アプリケーション分野で共通に利用できる AGT, API

# SWA-AF

## アプリケーションイネーブラー事例

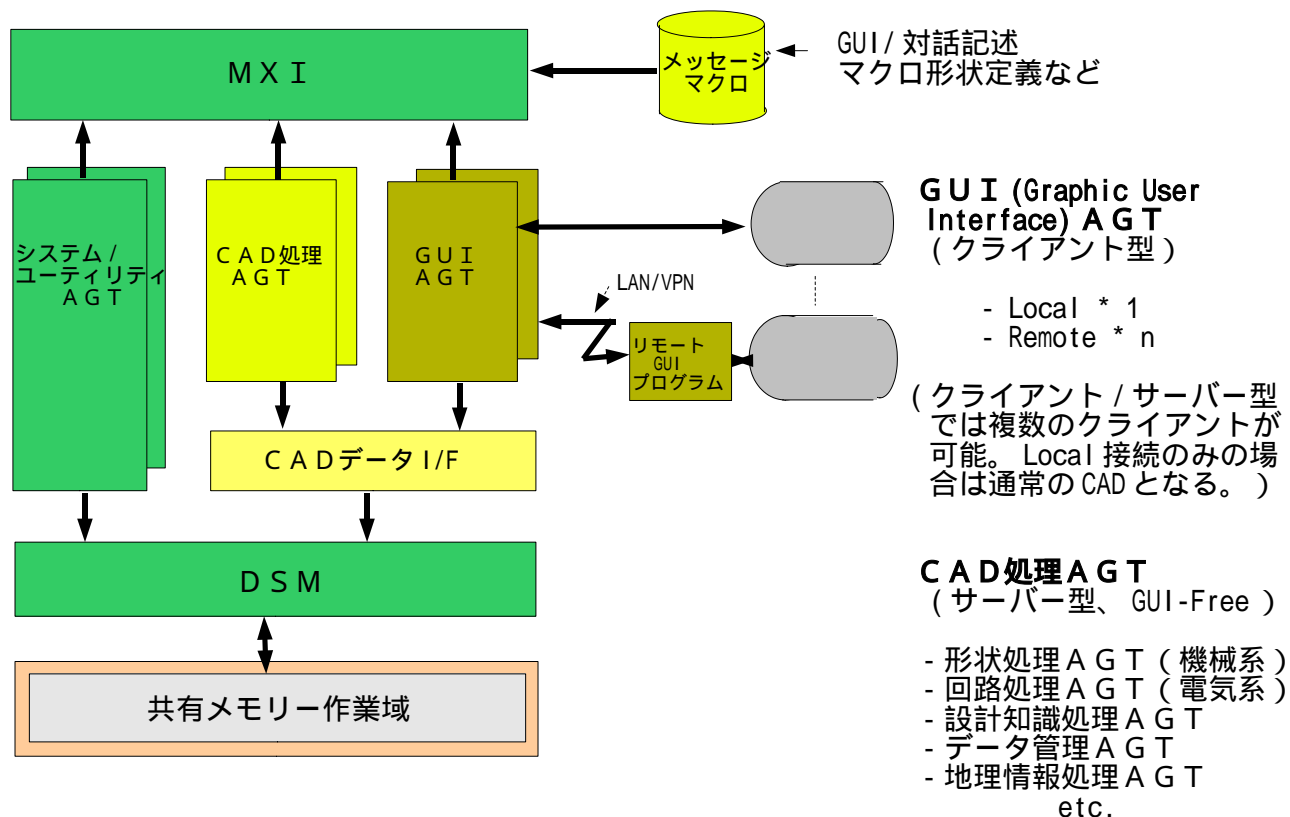
### ・CADイネーブラー



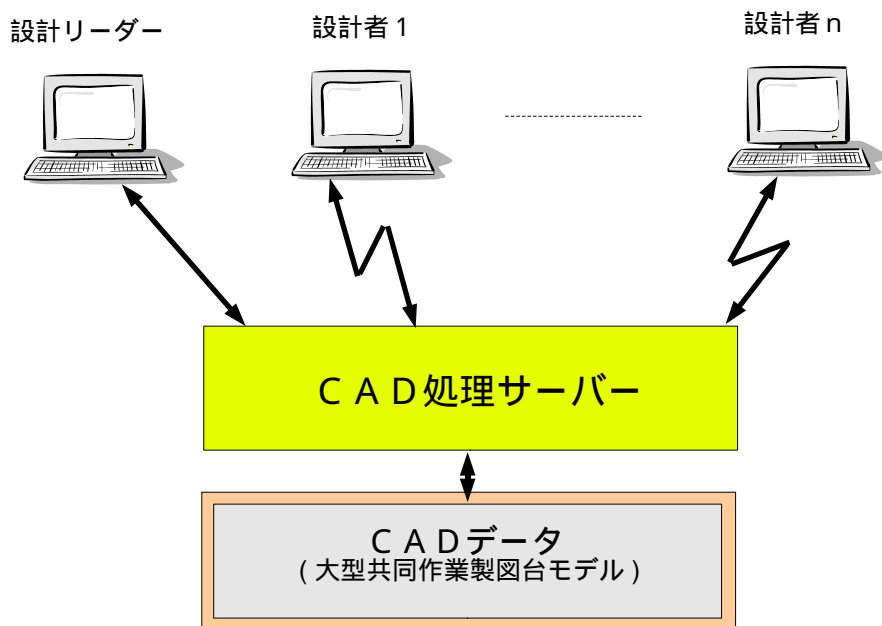
# SWA-AF 適用例 (1)

## - クライアント / サーバー型の活用

CADシステム (マルチクライアント共同設計CAD)



## 一つの設計対象を複数の設計者で同時共同設計



- サーバーの作業域でデータを集中・一元保持

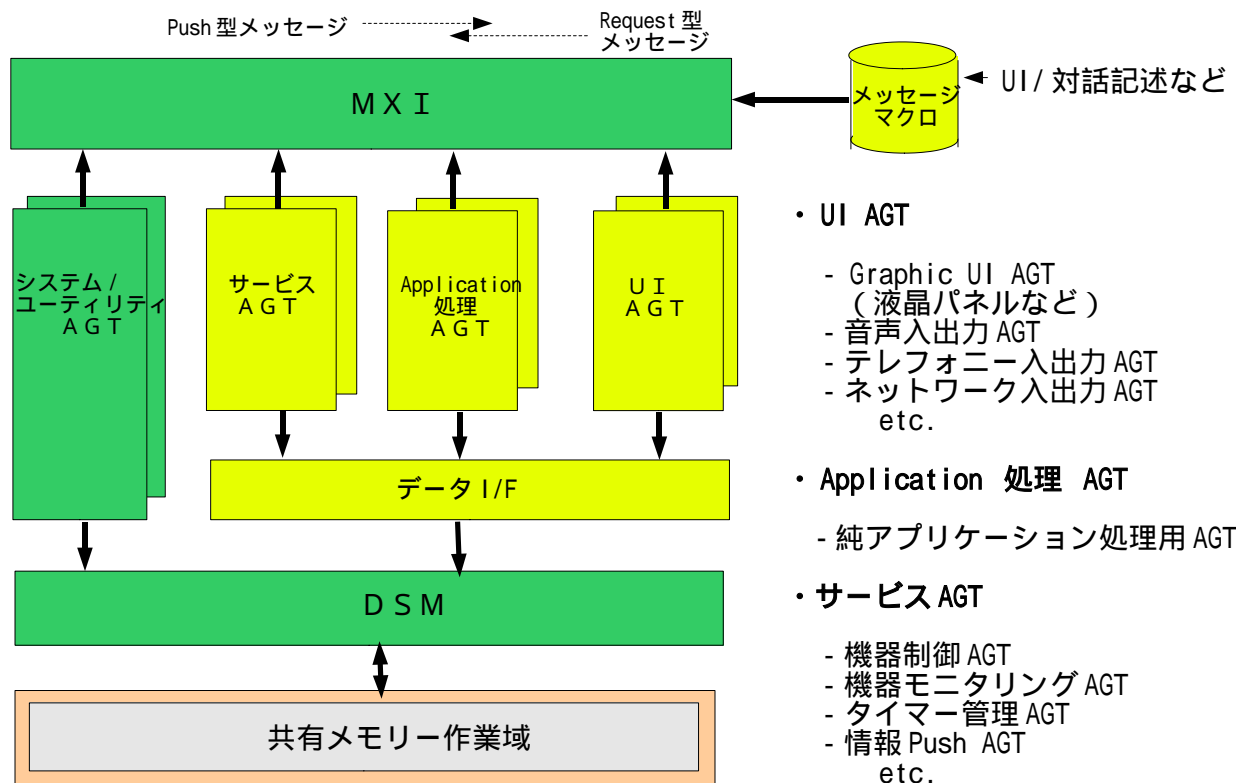
- ・リアルタイムコラボレーション
- ・設計者間のデータの不一致の心配がない
- ・設計途中のデータがリモートに散逸することなく  
管理上、セキュリティ上優位



## SWA-AF 適用例 ( 2 )

### - P u s h 機能 + クライアント / サーバー型の活用

#### マルチ UI + 機器制御システム



## 適用分野

- ネット家電集中制御ホームステーション
- デジタル家電組み込み制御システム
- 車載ステーション

### (例) NAVI システム

- UI AGT
  - GUI AGT (対話および各種表示)
  - 音声入力 AGT
  - 音声出力 AGT
  - リモコン入力受付 AGT
- Application 処理 AGT (可能な限り UI-Free のスタイルにする)
  - 目的地設定 AGT
  - 経路検索・経路設定 AGT
  - イベント検知 AGT (交差点接近、経路はずれなど)
  - 地図情報検索 AGT (ガソリンスタンド、コンビニなど)
  - 地図データ先読み AGT
  - その他特化アプリケーション AGT
- サービス AGT
  - GPS 位置取得 AGT
  - 車両状況取得 AGT
  - 道路状況取得 AGT
- 共有メモリー作業域
  - 現在位置データ、地図データ CACHE、目的地・経路データなど