



# *OpenSESSAME Seminar*

組込みソフトウェア技術者・管理者向けセミナー

初級者向けテキスト

組込みソフトウェア管理者・技術者育成研究会  
- SESSAME -

(<http://www.sesame.jp/>)

\*\*\*\*\* 目 次 \*\*\*\*\*

1 . SESSAME の紹介およびコースの概要	1
2 . 開発課題と失敗事例の解説	4
3 . 組込み向け構造化分析の例・設計の概要(1)	21
4 . 組込み向け構造化分析の例・設計の概要(2) 実習/回答と補足説明	44
5 . 組込み向け構造化設計(1)	51
6 . 組込み向け構造化設計(2) 実習/回答と補足説明	70
7 . プログラミング 組込み用語基礎知識	73
8 . ソフトウェアテストの概要	109
9 . プログラミング実習への説明	139
10 . プログラミング 実習	149
11 . プログラミング 実習の回答と補足説明	161
12 . ソフトウェアテスト 実習	164
13 . ソフトウェアテスト 実習/回答と補足説明	166
付録 . 話題沸騰ポットのシミュレーション	174

\*\*\*\*\*

## 組込み向け構造化設計 (1)

山田 大介



1. SESSAMEの紹介およびコースの概要
2. 開発課題と失敗事例の解説
3. 組込み向け構造化分析の例・設計の概要(1)
4. 組込み向け構造化分析の例・設計の概要(2)  
実習/回答と補足説明
5. 組込み向け構造化設計(1)
6. 組込み向け構造化設計(2) 実習/回答と補足説明
  
7. プログラミング 組込み用語基礎知識
8. ソフトウェアテストの概要
9. プログラミング実習への説明
10. プログラミング 実習
11. プログラミング 実習の回答と補足説明
12. ソフトウェアテスト 実習
13. ソフトウェアテスト 実習/回答と補足説明

付録・話題沸騰ボットのシミュレーション

101

## アジェンダ

1. 構造化設計
  - 分析モデルから設計の構造図へ
2. 設計の品質
  - モジュール分割、凝集度と結合度
3. アーキテクチャ設計
  - 非機能要件の設計
4. モジュール仕様
  - 構造図からモジュール仕様へ
5. 仕様変更への対応
  - 長持ちする設計

102

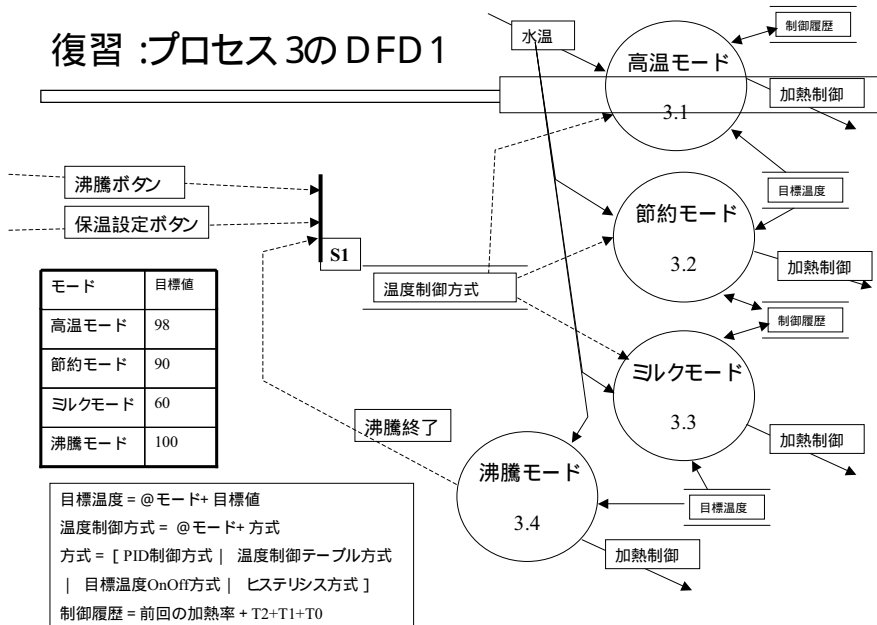
## アジェンダ

1. 構造化設計
2. 設計の品質
3. アーキテクチャ設計
4. モジュール仕様
5. 仕様変更への対応

103

SESSAME CONTENTS 2004

## 復習 :プロセス3のDFD1



104

SESSAME CONTENTS 2004

## 構造化設計の手順

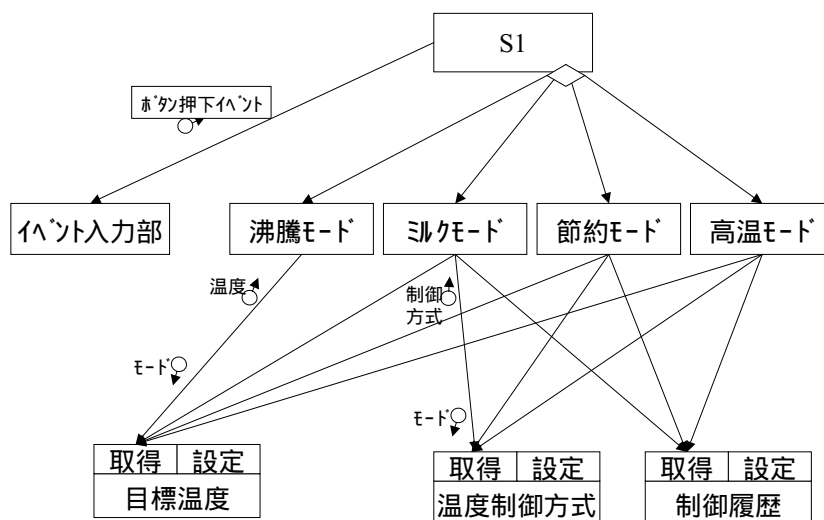
- 構造化分析のDFDに基づき、
- 最大抽象点を見つけて、源泉 - 変換 - 吸収スタイルを作る
  - その際に、ボスとして制御モジュールを作る
- 制御バーをモジュールに置き換える
  - ディスパッチャになる
- その他のバブルをモジュールに置き換える
  - そのバブルの下に、下の階層のバブルがぶら下がっていく
- 外部データの入力モジュールを追加する
- データの入出力を考える
  - データのアクセスにはデータ隠蔽モジュール

次に示す 2つの案が簡単に設計できる

105

SESSAME CONTENTS 2004

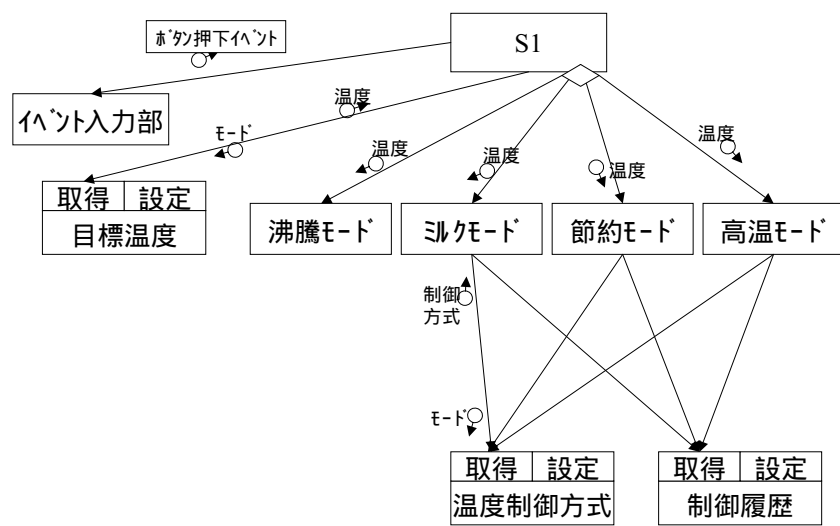
## 案 1



106

SESSAME CONTENTS 2004

## 案 2



107

SESSAME CONTENTS 2004

## 構造図の要点

- 構造図は、分析時のDFDからある程度機械的に作成できる
  - 最大抽象点を見つけることにテクニックは必要
    - 最も本質的な形で入力表現しているデータフローと、最も本質的な形で出力表現しているデータフローで囲まれた部分
- あとは、次に示す設計基準 (凝集度と結合度) により、モデルの選択と、若干の構造見直しを行う

108

SESSAME CONTENTS 2004

## アジェンダ

---

1. 構造化設計
- 2. 設計の品質
3. アーキテクチャ設計
4. モジュール仕様
5. 仕様変更への対応

109

SESSAME CONTENTS 2004

## 設計の品質

---

- モジュール分割
- モジュールの凝集度 (コヒージョン)
- モジュール間の結合度 (カップリング)

110

SESSAME CONTENTS 2004

## モジュール分割

---

- モジュールサイズの適正化
  - 1ページの半分程度 (30行)
  - 実装が単純になる
- システムの明解さ
  - 均衡型システム (左右均衡、上下は論理 - 物理)
  - モジュールの関係だけでシステムが理解できる
- 重複の極小化
  - 同じ機能を複数のモジュールに持たせない
- 情報隠蔽
  - モジュールで取り扱うデータを隠蔽する

111

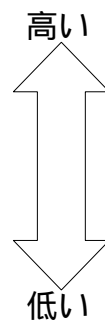
SESSAME CONTENTS 2004

## 凝集度 (コヒージョン)

---

- 凝集度は「高い」ほど保守性が良い

1. 機能的
2. 逐次的
3. 通信的
4. 手順的
5. 一時的
6. 論理的
7. 偶発的



112

SESSAME CONTENTS 2004



## 良い凝集度

---

1. 機能的
  - 単一の目的を持った機能、情報、責務。
2. 逐次的
  - 同一のデータに対する、
  - 関連が強く必然的な順番を有す機能集合。
3. 通信的
  - 同一のデータに対する、
  - 複数目的の機能集合。

**以上が、凝集度が高く保守しやすいモジュール**

113

SESSAME CONTENTS 2004

## 中程度の凝集度 ~ 悪い凝集度

---

4. 手順的
  - 関連の薄い複数データに対する、順次機能。
5. 一時的
  - 同一時間に発生する、互いに関係のない機能集合。
6. 論理的
  - 外部から利用する時に便利な機能の寄せ集め。
7. 偶発的
  - まったく相関のない機能の寄せ集め。
  - 実装制約 (HOW項目) を意識してしまうと、、

**時の経過とともに、  
モジュールの存在理由が曖昧になっていく**

114

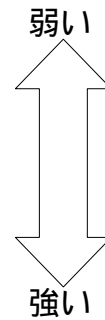
SESSAME CONTENTS 2004

## 結合度 (カップリング)

---

- 結合度は「弱い」ほど保守性が良い

1. データ結合
2. 構造体結合
3. バンドリング結合
4. 制御結合
5. ハイブリット結合
6. 共通結合
7. 内容結合



115

SESSAME CONTENTS 2004

## 良い結合度

---

1. データ結合
  - 構造を持たない単一のデータ。
2. 構造体結合
  - 同一の目的を持つデータメンバの集合。
3. バンドリング結合
  - 複数のフィールドから構成される構造体。
  - 使い方自由の万能構造体は良くない。

**以上が、結合度が弱く**

**理解しやすく、波及作用が限定的なモジュール間結合**

116

SESSAME CONTENTS 2004

## 中程度の結合度 ~ 悪い結合度

---

4. 制御結合
  - 相手のモジュールを制御するフラグ。
5. ハイブリット結合
  - 値の範囲により意味の異なるデータやフラグ。
6. 共通結合
  - グローバルデータ。
7. 内容結合
  - 相手のモジュールの内部を参照。(アセンブラ)

**変更すると、思わぬところに副作用が、、、**

117

SESSAME CONTENTS 2004

## 分析の効用

---

分析を行うことにより

- 要求の曖昧個所や矛盾点に気づくだけでなく
- ある程度機械的に設計が可能となる
- さらに、「凝集度」が高く、「結合度」が弱い、という高品質な設計も手に入る

**上流工程で品質を作り込もう**

118

SESSAME CONTENTS 2004

## アジェンダ

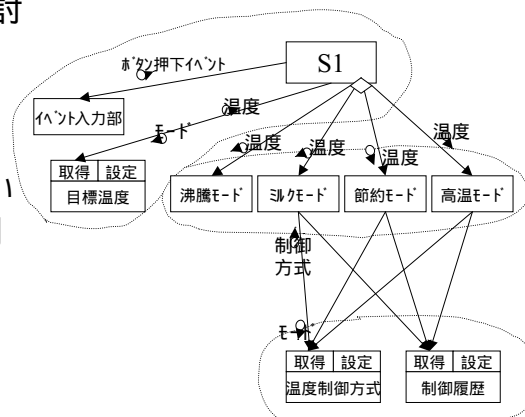
1. 構造化設計
2. 設計の品質
3. アーキテクチャ設計
4. モジュール仕様
5. 仕様変更への対応

119

SESSAME CONTENTS 2004

## アーキテクチャ設計

- システム外部からのイベント
  - 制御スレッドの検討
  - 割り込みの利用
- 並行処理
  - パッケージ分割
    - 構造図で破線囲い
  - サブシステム分割
    - 人員配置にも



120

SESSAME CONTENTS 2004

## 非機能要件

- 時間的制約
  - 応答時間
    - ハードリアルタイム / ソフトリアルタイム
- リソースの制約
  - ROM / RAM
  - CPUパワー
- 環境上の制約
  - 開発環境、工場検査、市場メンテナンス
- フォールトトレランス
  - 過負荷、フェイルセーフ

121

SESSAME CONTENTS 2004

## 非機能要件 :ポットの例

- レートモニタリング要求
  - ヒータのOnOff制御 (PID制御は、1秒毎)  
精度 ± 2.0mSecラインクロック
- イベントドリブン要求
  - 蓋の開閉検出 必須
  - 温度異常検出 不用
  - 満水と水なし検出 不用ハート割り込み
- データ構造要求
  - 温度履歴の保持方法3点までのリニアバッファ invocation に対応
- 復帰のメカニズム
- 製品対応要求
  - ROM実装量を最小化 + 保守容易化SCM\*環境で開発
- エラー処理メカニズム要求
  - エンド月|数チェックとトラップメカニズムコーディングルールに対応

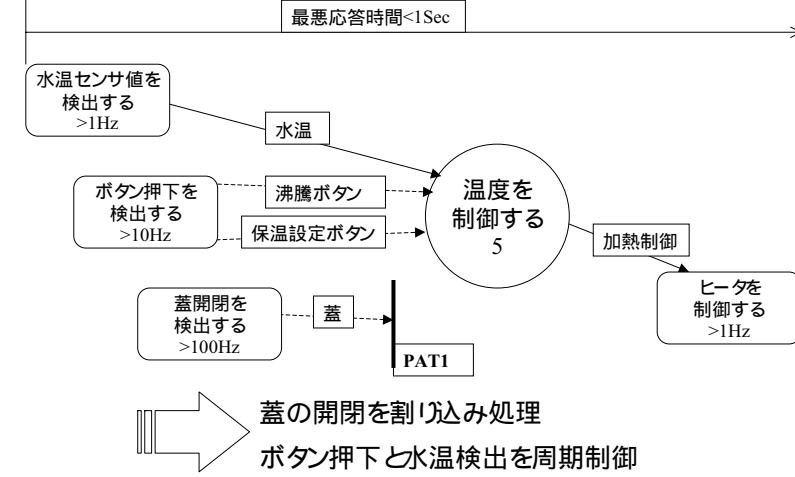
\*SCMとは、ソフトウェア構成管理

122

SESSAME CONTENTS 2004

## 非機能要件 ポットの例

### 「温度を制御する」の深いIDFD

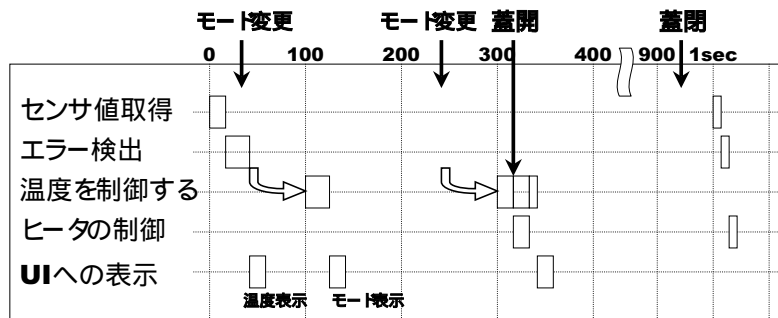


123

SESSAME CONTENTS 2004

## タイミングと並行性の検討

- センサ値取得が、1sec周期
- ボタン押下 (モード変更) の検出が、100msec周期
- 蓋の開閉は、割り込み処理



124

SESSAME CONTENTS 2004

## タイミングと並行性の検討

---

- タイムリー性
  - ワーストケースを積み上げてみる
  - ハードリアルタイム      ハードウェア処理の遅延がないか？
  - ソフトリアルタイム      ユーザへの応答性は大丈夫か？
- 並行性
  - 処理時間がかかるもの、ウエイトが発生するもの、の抽出
  - スケジューリング方式を検討
    - イベントドリブン、レートモニタック
  - 共有するリソースやデータが存在するか？
    - 何らかの排他処理

125

SESSAME CONTENTS 2004

## ここあたりに経験に基づく「暗黙の知識」が？

---

- スケジューリング方式の選定
  - パフォーマンスチューニング
  - ROMサイズの最適化
  - RAMのセクション設定、スタック量の見積もり
  - フォールトトレランス
- など？

皆さんが普段留意している事項を書きとめてください。  
可能であれば失敗事例もお願いします。

126

SESSAME CONTENTS 2004

## アジェンダ

---

1. 構造化設計
2. 設計の品質
3. アーキテクチャ設計
- 4. モジュール仕様
5. 仕様変更への対応

127

SESSAME CONTENTS 2004

## モジュール仕様

---

- モジュール単位のインターフェイス仕様と機能仕様
- インターフェイス仕様
  - そのモジュールが「何をすべきか」をインターフェイス仕様で定義
  - インターフェイス契約
- 機能仕様
  - そのモジュールが「どのように実現するか」を擬似コードで定義
  - 分析時のP-SPECを利用可能

128

SESSAME CONTENTS 2004



## モジュール仕様一覧の例

---

- 割り込み処理
  - 蓋開で、ヒーター制御を停止
- 制御クロック算出
  - ラインクロックでの制御周期計算
- 「温度を制御する」のディスパッチャ
- 「温度を制御する」のイベント入力部
- 「温度を制御する」の高温モード
- 「温度を制御する」のミレクモード
- 「温度を制御する」の沸騰モード

129

SESSAME CONTENTS 2004

## モジュールのインターフェイス仕様の例

---

名 称 : 温度を制御する」の高温モード  
目 的 : 高温で保温するときの温度制御  
引 数 : なし  
戻り値 : なし

130

SESSAME CONTENTS 2004

## モジュールの機能仕様の例

### • 擬似コード – P-Specより抜粋

制御周期ごとに、

1. //水温履歴を更新する

$T_2 = T_1$

$T_1 = T_0$

$T_0 =$  水温

2. //目標温度を得る

$T_g$ を目標温度 .モード=高温モードである  
目標温度 ,目標値に設定する

3. //PIDでの制御量を計算する

$M = K_p(T_1 - T_0) + K_i(T_g - T_0) + K_d(2T_1 - T_0 - T_2)$

4. //ヒータ通電期間を制御する

制御期間 × 今回の加熱率 の期間中は、  
加熱制御="on"

この期間が終了するタイミングで  
ここにはちがった設定が、、、

加熱制御="off"

//加熱率の履歴を更新する

前回の加熱率 = 今回の加熱率 //更新

131

SESSAME CONTENTS 2004

## アジェンダ

1. 構造化設計
2. 設計の品質
3. アーキテクチャ設計
4. モジュール仕様
5. 仕様変更への対応

132

SESSAME CONTENTS 2004

## 仕様変更 ポットの例、例えば、

- ミレクモードのときは、「62度を上限値、58度を下限値」という仕様変更
- いきなりプログラミングした場合、
  - プログラムの流れを順を追って読み、
  - ソースコード上で変更箇所を特定する
  - スパゲッティになってしまっている場合は、
  - 変更箇所の特定すらできないことも
  - また変更による副作用も恐い

62度を超えたぞ、  
ミレクモードだから、  
一度沸騰させているし、  
ヒータのエラーにもなっていないから、  
ここでヒータを停止させよう  
でもこの手前の分岐は何かかな？  
今ヒータ止めてもいいのかな？
- 設計図がある場合、
  - 構造図から変更モジュールが特定できる
  - 外部のインターフェイスを変えないように、内部の処理を変えよう

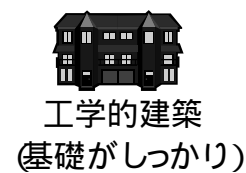
➡ **変更箇所の特定は素早く、修正は局所的**

133

SESSAME CONTENTS 2004

## 仕様変更への対応

- プログラムの流れを追って修正 (増築)
  - ソースコードを見ないと修正箇所が特定できない
  - やがてスパゲッティ化
  - ソースコードだけが信頼できる
- 設計図から修正箇所を特定
  - 修正箇所が局所化される
  - 制御された修正



134

SESSAME CONTENTS 2004

## 仕様変更への対応

---

- 「長持ちする設計」が良い設計
  - 固定変動分離
    - 外部へのインターフェースを固定
    - 内部処理は変動容易
    - 固定部から変動部を呼び出して、変動部だけを修正
- 設計図でレビューしましょう
  - ソースコードを追わないと分からない
    - 作った人しか分からない
    - さらに、副作用に関しては未知
  - 抽象度の高い設計図でのレビューが効果的

135

SESSAME CONTENTS 2004

## まとめ

---

1. 構造化設計
  - 分析さえしっかりすれば設計は簡単。分析しましょう!
2. 設計の品質
  - 高い凝集度と弱い結合度で、保守しやすいソフトウェアへ!
3. アーキテクチャ設計
  - いろいろ暗黙の知識がありそうですね。
  - 失敗事例などの知識を共有化する仕組みを作りましょう!
4. モジュール仕様
  - 構造図さえあればここも簡単。但し、状態やモードに注意。
5. 仕様変更への対応
  - 「長持ちする設計」を目指し、設計図でレビューしましょう!

136

SESSAME CONTENTS 2004

以上が、設計でした。  
次はもちろん (やっど?) プログラミングです。

その設計は、変更能耐えられますか?

## 参考

---

- Meilir Page-Jones 『構造化システム設計への実践的ガイド』, 近代科学社, 1991

## 本ドキュメントのご利用に際して

- 本著作物の著作権は作成者または作成者の所属する組織が所有し、著作権法によって保護されています
- SESSAME は本著作物に関して著作者から著作物の利用 を許諾されています
- 本著作物は SESSAME が利用者個人に対して使用許諾を与え、使用を認めています
- SESSAME から使用許諾を与えられた個人以外の方で本著作物を使用したい場合は [query@sessame.jp](mailto:query@sessame.jp) までお問い合わせください

SESSAME が著作者から許諾されている権利

著作物の複製・上演・演奏・公衆送信及び送信可能化・口述・展示・上映及び頒布・貸与・翻訳・翻案・二次的著作物の利用

- ドキュメント中には Microsoft 社, Adobe 社等が著作権を所有しているクリップアートが含まれています

---

### OpenSESSAME Seminar

#### 組込みソフトウェア技術者・管理者向けセミナー 初級者向けテキスト

---

2002 年 10 月 15 日 初版 第 1 刷発行

2003 年 10 月 29 日 初版 第 2 刷発行

2004 年 3 月 19 日 第 2 版 第 1 刷発行

2004 年 4 月 30 日 第 3 版 第 1 刷発行

2004 年 6 月 17 日 第 4 版 第 1 刷発行

**著 者** 上原慶子、大野晋、坂本直史、鈴木圭一、須田泉、西康晴、  
二上貴夫、三浦元、三宅貴章、森孝夫、山田大介、山崎辰雄

**編集・発行** 組込みソフトウェア管理者・技術者育成研究会  
(SESSAME)

<http://www.sessame.jp>