

# ソフトウェアテストの前に

## Agenda

- テストの前に
- ソフトウェアの社会的役割
- ソフトウェアとは
- ソフトウェアを理解する

## テストの前に

- テスト対象、すなわちソフトウェアとはどういうものなのか？ を理解する
  - 「敵を知り、己をしれば、百戦危うからず」  
テスト対象は敵ではありませんが...
- そのために
  - まず、より上位の視点からソフトウェアというものを俯瞰
  - 次に、例を挙げて紐解いてみましょう
    - 電気やかんのソフトウェア
    - ... 要求仕様を実現する制御ソフトウェア

3

## ソフトウェアの社会的役割

- 提供するソフトウェア(ハードウェア制御を含む)でサービスを確実に実現する
  - 銀行のATM
    - 預け入れ、引き出し、振り込み等のサービスが間違いなく行えること
    - 他人に個人の情報が渡らず安心して使えること
    - ...
  - 携帯電話
    - 圏内では音声通話、データ通信が途切れることなく行えること
    - 電池消費量が適正であること
    - ...



4

## ソフトウェアの社会的役割

- もし、ソフトウェアに不具合があると？
  - － 銀行のATM
    - 必要な手続きができず、取引ができない
    - セキュリティに問題があると、顧客資産の流出の可能性も
  - － 携帯電話
    - 使えない電話としてキャリア／メーカーへのダメージ
    - 緊急時に使えない場合には人命に関わることも
  - － 自動車関連
    - 電子制御の不具合で暴走やブレーキ不良による事故の可能性
  - － 証券取引システム
    - 取引の停止や不整合により経済活動へ大きな混乱をきたす
  - － ライフライン
    - 電気、水道、ガスの制御システムに不具合があると、産業界だけでなく生活全体に大きな影響を及ぼす



5

## ソフトウェアの社会的役割

- いまや、社会生活を送るうえでソフトウェアの役割は非常に大きくなっている



**「ソフトウェアの品質確保」は重要事項**

**「ソフトウェアの品質確保」とは言うけど、  
そもそも「ソフトウェア」って何なの？**

6

## ソフトウェアとは

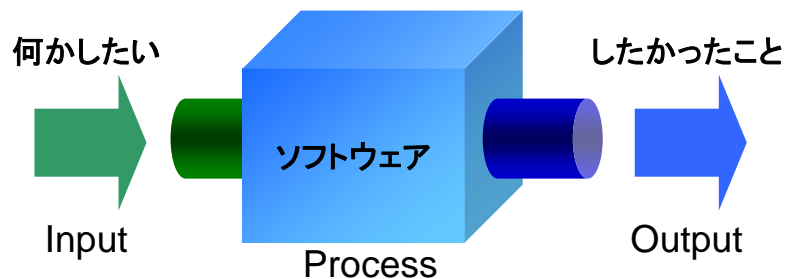
- JIS X 0001 によると
  - 情報処理システムのプログラム、手続き、規則  
および関連文書の全体又は一部分  
(備考)ソフトウェアは、それを記憶した  
媒体とは無関係な知的創作物である
- なぜ「ソフトウェア」を作るのか？
  - 仕事だから？ 趣味だから？
  - 共通して言えることは  
「ソフトウェアを実行することにより何かを得たい」  
から



7

## ソフトウェアとは

- 何かしたいことに必要な情報をソフトウェアに  
与え (Input)、ソフトウェアで何か処理を行い  
(Process)、したかった結果を得る (Output)
- Input - Process - Output (IPOモデル)  
が成立



8

## ソフトウェアを理解する

- ソフトウェアを理解するには、この  
Input - Process - Output  
のモデルを使って紐解く、すなわち分析する  
ことが効果的
  - モデルを使うことにより可視化でき、対象ソフトウェアについての認識合わせにも役立つ
- 電気やかんのソフトウェアをモデルを使って  
分析する・・・グループワーク2
  - 今回はセンサー情報等の「データ」の流れを表現しやすい「データ・フロー・ダイアグラム (DFD)」を使用

## グループワーク2

## Agenda

- グループワーク2解説
  - お題
  - DFD
  - DFD を使った分析
  - コンテキスト・ダイアグラム
  - DFD 第1層
- グループワーク2
- まとめ

11

## お題

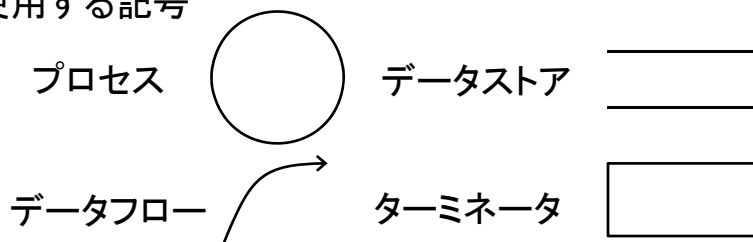
- 対象システムである”電気やかん”の一部「水温を制御する部分」のDFDを作成する
  - ワーク1のテスト対象部分を分析する
  - 入力と処理(変換)と出力に着目
    - 入力データを処理して出力データを得る
      - ソフトウェア設計者には馴染みやすい考え方
    - 入力データを与えて出力データを確認する
      - テスト設計と実施にも馴染みやすい考え方



12

## DFD

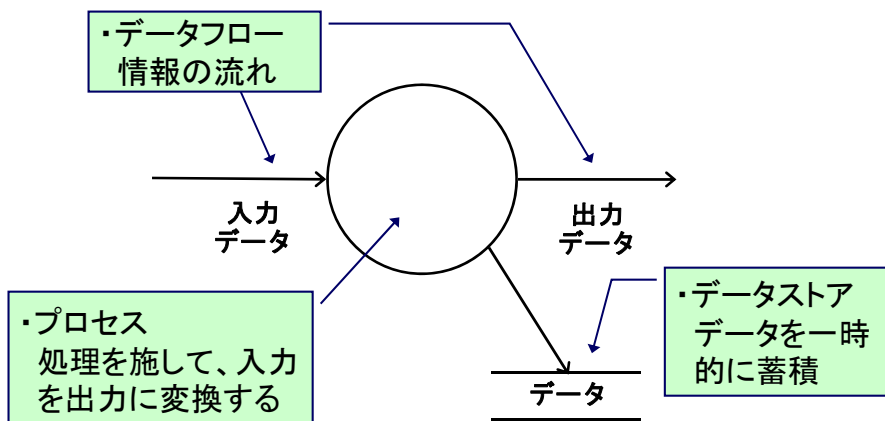
- 対象システムの機能要求を、データフロー(入力、出力)とプロセス(処理)のネットワーク図として表したもの
- 入力データに対して何らかの処理を施して出力データを得る  
これにより、データの流れ、データがどのように変わっていくかに注目して、対象システムを実現するために必要な要件を考察する
- 使用する記号



13

## DFD

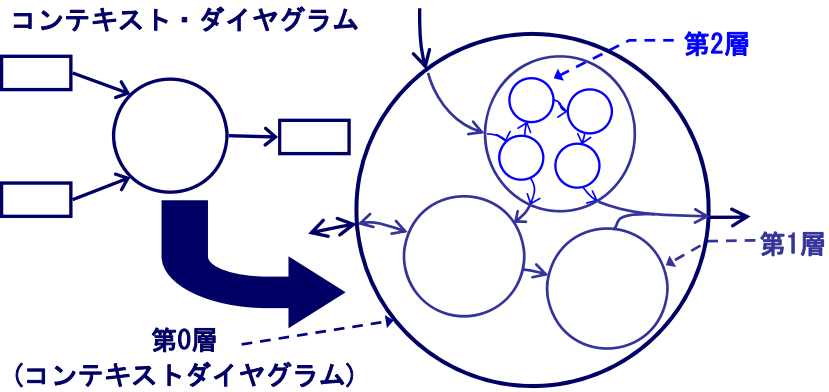
- ソフトウェアをデータの処理(変換)ととらえる



14

## DFD を使った分析

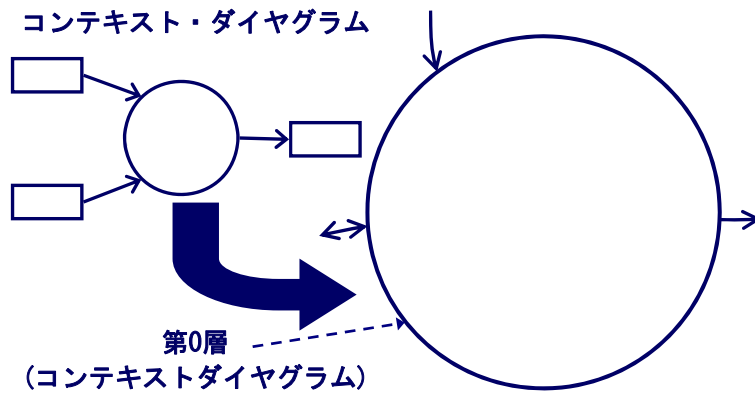
- 入力と処理と出力に着目
  - トップダウン、階層構造化、段階的詳細化



15

## DFD を使った分析

- コンテキスト・ダイアグラム
  - トップダウン、階層構造化、段階的詳細化

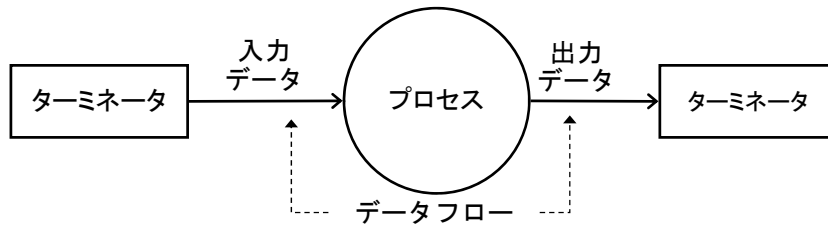


16



## コンテキスト・ダイアグラム

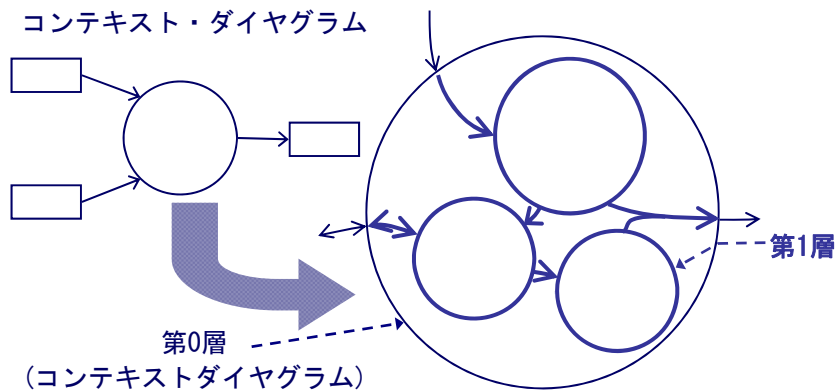
- 最上位のDFD = 対象システムを俯瞰
  - 対象システムにおいて考慮すべき外部要件
  - 対象システムと外部環境の間にあるデータの流れ
- 今回
  - プロセス: 電気やかんのソフトウェア
  - ターミネータ: 電気やかんのソフトウェアの外にありデータをやりとりする



17

## DFD を使った分析

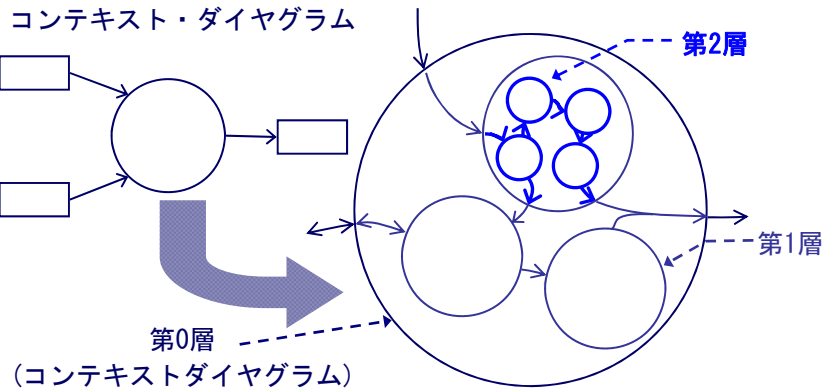
- 第1層に分解
  - トップダウン、階層構造化、段階的詳細化



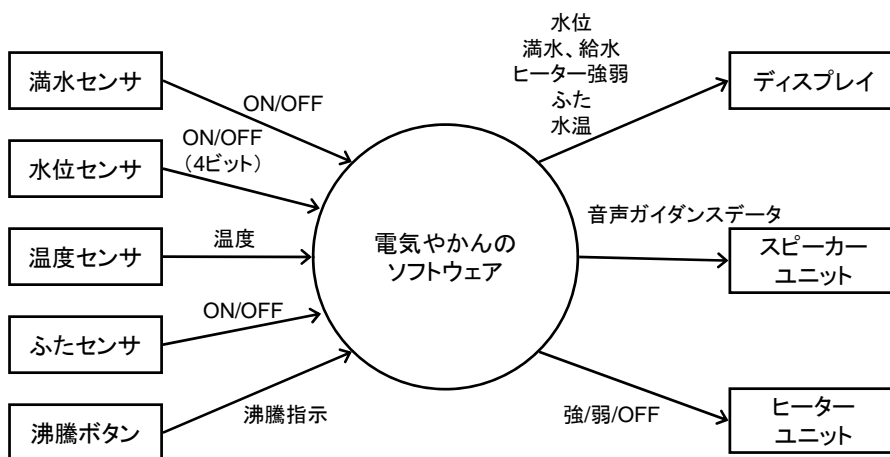
18

## DFD を使った分析

- 第2層に分解
  - トップダウン、階層構造化、段階的詳細化



## コンテキスト・ダイアグラム



ターミネータや入出力データの名称は、仕様書と対比して理解しやすいように、より具体的な表現にしています

## DFD 第1層

- コンテキスト・ダイアグラムのプロセスを主要機能にブレークダウンする
- 電気やかんの要求仕様書の機能一覧

機能名	内容
湯沸かし機能	やかん形状の筐体に蓄えられた水を安全にお湯にする。
安全確認機能	湯沸し機能の実行前、実行中に安全を確認し、安全が確認できない場合は、湯沸し機能の実行を抑制する。
表示機能	やかん形状の筐体に蓄えられた水の状況を専用の表示システムにて使用者に通知する。 通知する状況は下記の通りである。 ・フタの開閉状態 ・水位 ・水温 音声での通知は、システムへの操作の開始・終了のタイミングで行う。

水温を制御する

事前条件を監視する

状態をユーザに知らせる

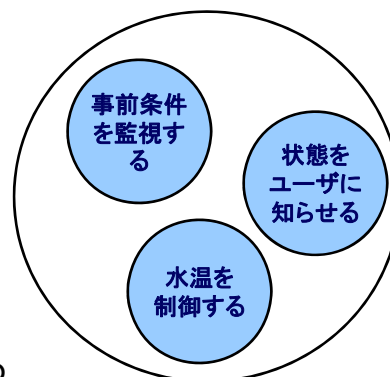
21

## DFD 第1層

- コンテキスト・ダイアグラムのプロセスを主要機能にブレークダウンする

- 例えば

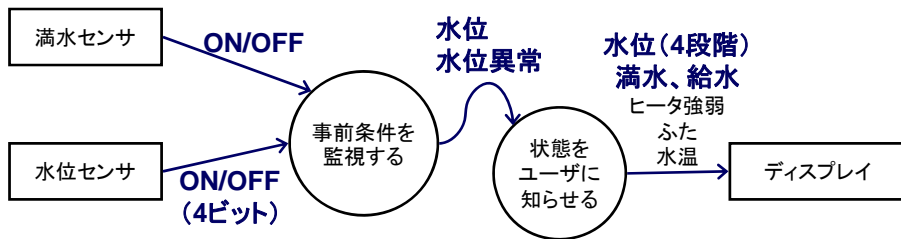
- 水温を制御する
  - 湯沸かし機能
- 事前条件を監視する
  - 安全確認機能
    - 水位(満水、水位)
    - ふた開閉
- 状態をユーザに知らせる
  - 表示機能



22

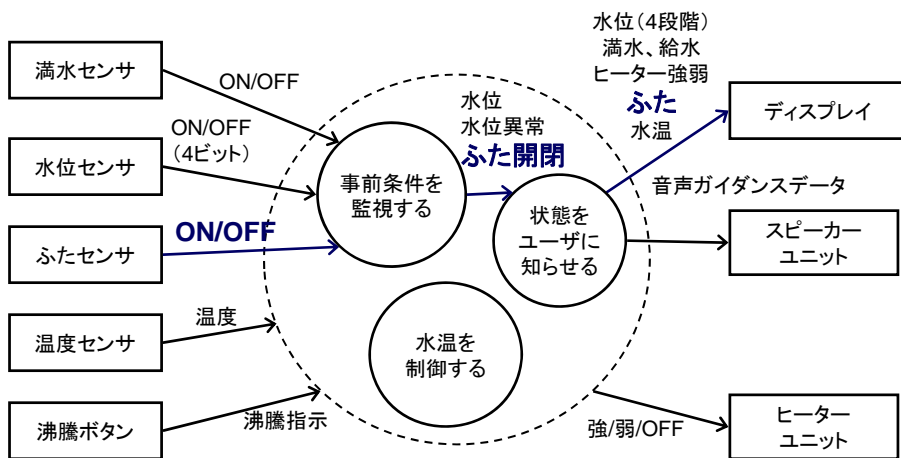
## DFD 第1層

- プロセスとターミネータ間、プロセス間の入出力フローを書き込む
- 例えば
  - 水位をディスプレイの水位メータに表示する
  - ディスプレイの満水、給水を表示する



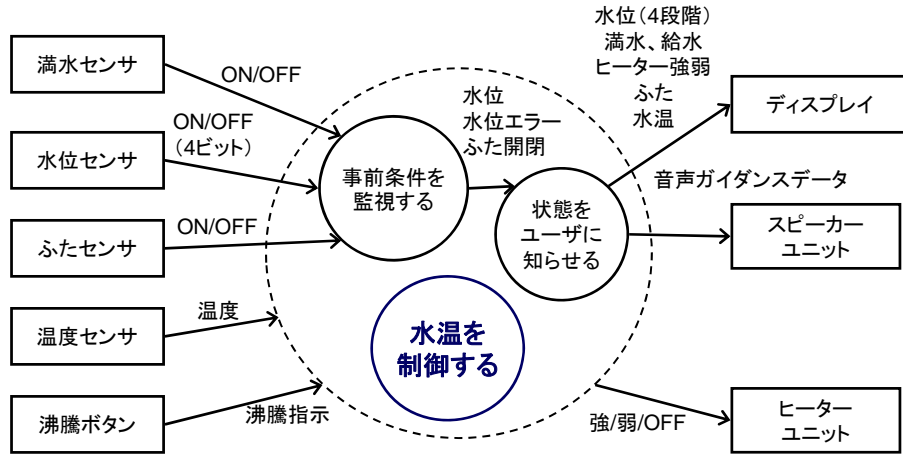
## DFD 第1層

- ふたの状態を表示する



## グループワーク2

### • ”水温を制御する”の流れ？

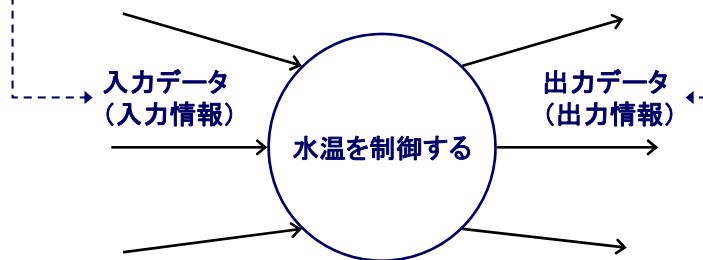


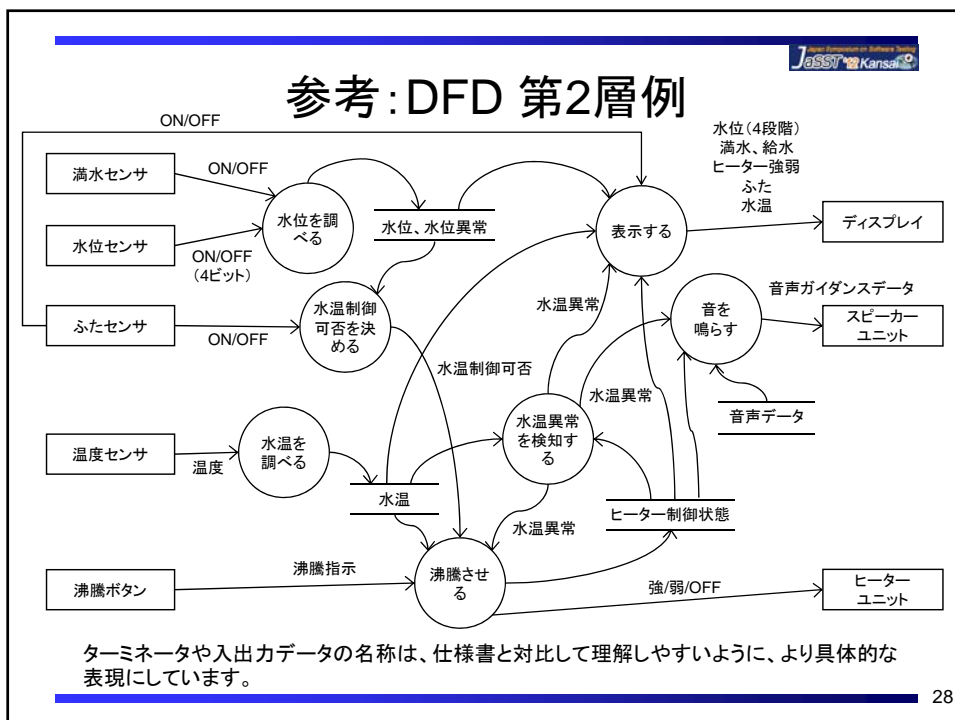
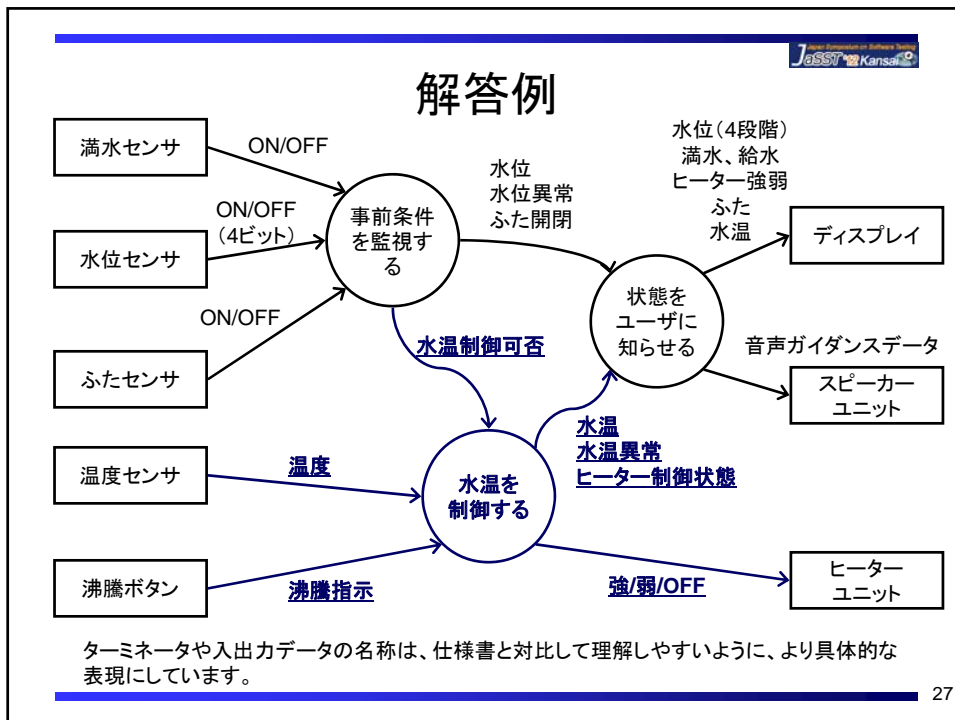
## グループワーク2

### • 作成してみましょう

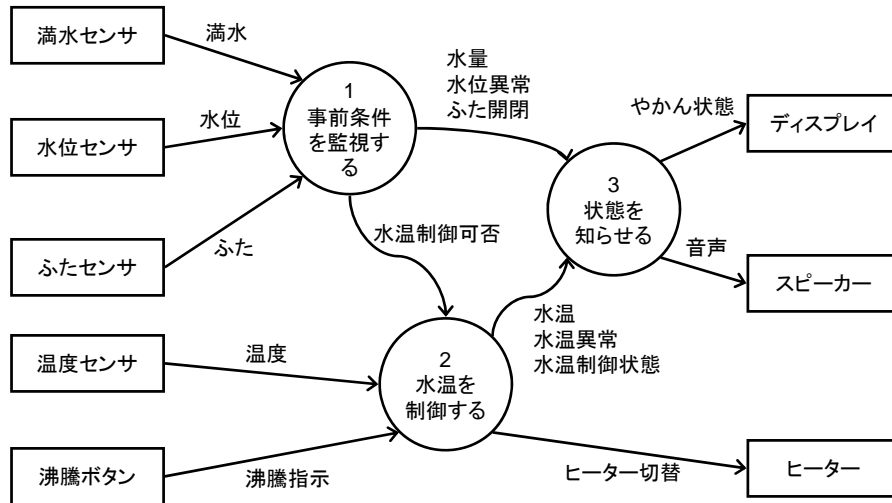
—あなたが”電気やかんのソフトウェア”の中の水温制御係だとしたら

- 誰からどんな情報をもらいますか？
- 誰にどんな情報を与えますか？





## 参考:DFD 第1層例



29

## 参考:データ辞書例

- 満水 = [ON | OFF] 「ON:満水、OFF:満水でない」
- 水位 = 整数 下限:0 上限:15 単位:なし  
「第1~4水位センサがbit0~3に対応する4ビットデータ」
- 水量 = 整数 下限:0 上限:5 単位:なし
- ふた = [ON | OFF]
- ふた開閉 = [閉 | 開]
- 温度 = 整数 下限:0 上限:151 単位:摂氏
- 水温 = 整数 下限:0 上限:110 単位:摂氏
- 水温異常 = [異常 | 正常]
- 水位異常 = [異常 | 正常]
- 水温制御可否 = [可 | 否]
- 沸騰指示 = [ON | OFF]
- ヒーター切替 = [強 | 弱 | OFF]
- 水温制御状態 = [沸騰開始 | 沸騰中 | 沸騰完了]+ヒーター切替
- やかん状態 = 温度+水量+ふた開閉+ヒーター制御+水位異常+温度異常
- 音声 = [まいど！ | いまからお湯を沸かしませ！  
沸いたで！！冷めんうちに使いいや！！ | 高温エラーです  
温度上がらずエラーです | 湯沸しが終わらずエラーです]

「」内は補足説明

30

## まとめ

- テスト対象、すなわちソフトウェアとはどういうものなのか、を理解する
  - 「敵を知り、己をしれば、百戦危うからず」  
テスト対象は敵ではありませんが...
  - 今回は対象システムの一部の仕様分析まで
- Input - Process - Output で紐解く
- テストへの展開
  - Input を与える
  - Output を観察する
  - 期待通りでないなら Process の欠陥を探す

