

# USDМを活用したスペックアウトとモデル設計

2025年5月23日  
株式会社 両毛システムズ  
松本 智也

# 会社概要

## RS 両毛システムズ

- 【設立】 1970年1月31日
- 【従業員数】 736名 (グループ全体1,007名)
- 【売上高】 181億円 (2024年3月期連結)
- 【顧客数】 約900社

### <その他オフィス>

- ・東京支社 (東京都)
- ・西日本サポートセンター (広島県広島市)
- ・中部サポートセンター (愛知県名古屋市、豊橋市)

仙台開発センター  
(宮城県仙台市)



本社  
(群馬県桐生市)



太田開発センター  
(群馬県太田市)

群馬県内最大規模の <システムインテグレータ>

# アジェンダ

---

- 概要
- 背景
- 直面した課題
- 課題解決へのアプローチ
- 課題に対しての結果
- 上手くいかなかったこと
- まとめ

# 概要

多機能なソースコードから基本機能を抽出し、Simulinkモデルへの移行を行いたい要望に対してUSDmを用いてソースコードをスペックアウトし、機能分割の再整理、モデル設計を行った事例です。

## 伝えたい人のイメージ

- ・既存のソースコードを理解し、機能分割し整理を行いたい人
- ・設計書情報が不足しているソースコードを更新する際にどこに手を付けてよいか困っている人

## 提示する価値

- ・ソースコード先行で進んだ開発で、トレーサビリティを確保したドキュメントを後から作成できる
- ・ソースコードで議論をするのではなく、階層化された機能構造を見える化し、検討できるようになる
- ・リバースで作成するSimulinkモデル設計のインプット情報を整理できる

# 背景

以下2つの要求を満たすための開発を実施

1. 複雑化した機能の整理を行いシンプルな機能だけ残して再構築
2. ソースコード開発からSimulinkモデル開発への移行

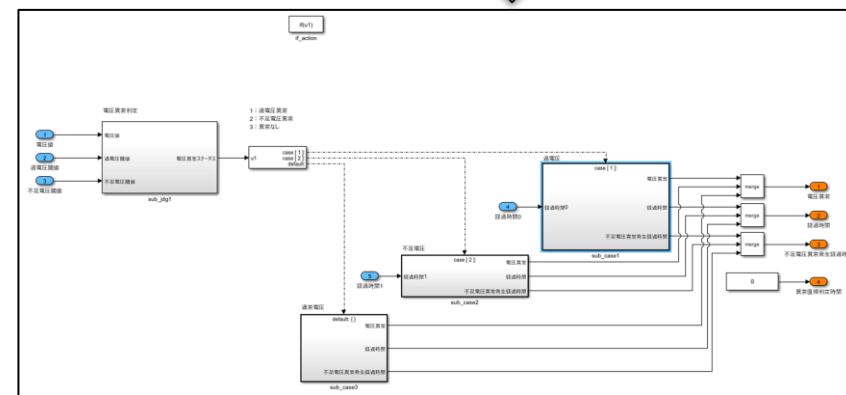
## 1. 機能整理の要望

既存製品から機能を整理、削除してシンプルな動作のみ行う状態へ再構築



## 2. モデル開発への移行

```
19 void check_over_voltage(uint16_t raw_voltage) {
20     /* 電圧値の有効範囲チェック */
21     if (raw_voltage > 4095) {
22         return VOLTAGE_OVER; // 異常値検出
23     }
24     /* 実際の電圧値に変換 (例: 5V基準の場合) */
25     uint32_t voltage_mv = (raw_voltage * 5000UL) / 4095;
26     /* 過電圧判定 */
27     if (voltage_mv > OVER_VOLTAGE_THRESHOLD) {
28         return VOLTAGE_OVER;
29     }
30 }
31 return VOLTAGE_NORMAL;
32 }
```



## 直面した課題

設計書、ソースコード間のトレーサビリティ情報の欠如により仕様理解が難解

ソースコードを読み込まないと話しに参加できない

現状のプロジェクトメンバーの中に対象製品開発に対する知識、経験がある人がいない

ソースコード開発からSimulinkモデル開発への移行経験がない

# 課題解決へのアプローチ

設計書、ソースコード間のトレーサビリティ情報の欠如により仕様理解が難解

ソースコードを読み込まないと話しに参加できない

現状のプロジェクトメンバーの中に対象製品開発に対する知識、経験がある人がいない

ソースコード開発からSimulinkモデル開発への移行経験がない

## 解決プロセスの流れ

ソースコード  
複雑な機能が混在した状態の既存コード

1

USDMを活用したスペックアウト

2

機能の分類と必要性の整理

3

階層構造を活かしたモデル設計

# 課題解決へのアプローチ

設計書、ソースコード間のトレーサビリティ情報の欠如により仕様理解が難解

ソースコードを読み込まないと話しに参加できない

現状のプロジェクトメンバーの中に対象製品開発に対する知識、経験がある人がいない

ソースコード開発からSimulinkモデル開発への移行経験がない

## 解決プロセスの流れ

ソースコード  
複雑な機能が混在した状態の既存コード

1

USDMを活用したスペックアウト

2

機能の分類と必要性の整理

3

階層構造を活かしたモデル設計



# 課題解決へのアプローチ ～USDMMを活用したスペックアウト

## 1. ソースコードの機能的な分類

- 変数名、コメント、処理内容から1機能のまとまりを推測し、グルーピング

## 2. 上位要求の抽出

- まとめた各グループが「何をするためのロジックか」を推測し、仮の上位要求として定義します。

## 3. 下位要求への分解

- 上位要求を実現するための具体的な振る舞いを、ソースコードの処理内容から抽出し、仮の下位要求として定義します。

## 4. 仕様への落とし込み

- 下位要求毎に処理を仕様として定義します。

```

VoltageStatus check_over_voltage(uint16_t raw_voltage) {
    /* 電圧値の有効範囲チェック */
    if (raw_voltage > 4095) {
        return VOLTAGE_OVER; /* 異常検出 */
    }

    /* 実際の電圧値に変換 (例: 5V基準の場合) */
    uint32_t voltage_mv = (raw_voltage * 5000UL) / 4095;
    /* 過電圧判定 */
    if (voltage_mv > OVER_VOLTAGE_THRESHOLD) {
        return VOLTAGE_OVER;
    }

    return VOLTAGE_NORMAL;
}

VoltageStatus check_voltage_status(uint16_t raw_voltage) {
    VoltageStatus status = check_over_voltage(raw_voltage);
    /* 既に異常状態の場合は即時返却 */
    if (status != VOLTAGE_NORMAL) {
        return status;
    }

    /* 低電圧判定用変換 */
    uint32_t voltage_mv = (raw_voltage * 5000UL) / 4095;

    if (voltage_mv < UNDER_VOLTAGE_THRESHOLD) {
        return VOLTAGE_UNDER;
    }

    return VOLTAGE_NORMAL;
}

```

要求	0002_0003_0000	各種センサーで異常値が計測された
理由		
説明		各種センサー値が異常値となっている
要求		xxx仕様の場合は電圧異常検知関連
理由		xxx仕様の場合は外部のICが管理する
説明		
機能無効化時		仕様
理由		
説明		
要求		電圧が製品の規定値を超えた場合
理由		製品仕様として15V以上がxx秒以上
説明		電圧が1.5V以下または1.2V以下
電圧異常判定 (xxxx.c 50行目)		仕様
理由		
説明		
過電圧異常検知		仕様
理由		
説明		
低電圧異常検知		仕様
理由		
説明		
正常時		仕様
理由		
説明		
過電圧 (xxxx.c 100行目)		仕様
理由		
説明		
ステータス遷移		仕様
理由		
説明		
動作停止通知		仕様
理由		
説明		
低電圧 (xxxx.c 200行目)		仕様
理由		
説明		
ステータス遷移		仕様
理由		
説明		
動作停止通知		仕様
理由		
説明		

ソースコードをグルーピング、仕様理解しながらUSDMM構造へまとめる

# 課題解決へのアプローチ ～USDMを活用したスペックアウト

## スペックアウト時に大事だと感じたポイント

例：製品の許容電圧を超えた場合は動作を停止したい場合

### NG例：ソースコードをそのまま転記

電圧チェック	要求	電圧が1.5V以上であればカウンタを進める	
	理由		
	説明		
	x x x 関数呼び出し	仕様理由	xxxx関数を使用して電圧を取得する
	電圧判定	仕様理由	取得した電圧が1.5V以上か判定する
	15V以上	仕様理由	異常電圧カウンタを1増加する
	15未満	仕様理由	異常電圧カウンタを初期化する

### OK例：下記ポイントを意識して作成

電圧異常確認	要求	電圧が製品の規定値を超えた場合は動作を停止する		閾値の根拠があって いるか確認
	理由	製品仕様として15V以上がxx秒以上継続は保証していないため		
	説明	電圧が1.5V以上がx x 秒以上継続しているか判定		
	電圧異常時間計測 (xxxx.c 50行目)	仕様理由	電圧が1.5V以上であればカウンタを進める	電圧値は電圧値監視機能の情報を 使用する(xxxx関数)
	電圧異常時間判定 (xxxx.c 100行目)	仕様理由	カウンタがxx秒以上となった場合は異常動作判定 機能に電圧異常を通知する	
		理由	電圧異常動作よりも優先度が高い異常があった場 合にそちらを優先するため	
		説明		

- ・作業的にソースコードをUSDM化するのではなく、意図を読み取った結果をUSDM化する  
要求(機能)と仕様(ロジック)の区別を意識して階層化する
- ・情報を埋め込みながら整理する  
USDM内に対応している関数、行数などの情報を記載する  
有識者に確認すべき内容について質問内容をコメントボックスで配置する

# 課題解決へのアプローチ

設計書、ソースコード間のトレーサビリティ情報の欠如により仕様理解が難解

ソースコードを読み込まないと話しに参加できない

現状のプロジェクトメンバーの中に対象製品開発に対する知識、経験がある人がいない

ソースコード開発からSimulinkモデル開発への移行経験がない

## 解決プロセスの流れ

ソースコード  
複雑な機能が混在した状態の既存コード

①

USDMを活用したスペックアウト

②

機能の分類と必要性の整理

③

階層構造を活かしたモデル設計

# 課題解決へのアプローチ ～機能の分類と必要性の整理

## 下記の内容について実施

- ・有識者と作成したUSDMの要求、理由が正しいかについて確認
- ・有識者と必要、不要、共通 機能の確認
- ・確認結果から変更点をまとめる
- ・構造の整理を実施

## 確認中の分類基準

必要な要求

不要な要求

変更が必要な要求

共通要求

		整理前	
異常判定機能	要求	0002_0003_0000	各種セ
	理由		各種センサ値が異常値となっている
	説明		各種センサ値が異常値となっている
	委小		xxx仕様の場合は電圧異常検知関連
	理由		xxx仕様の場合は外部のICが管理す
	説明		
	機能無効化時		仕様理由説明
	機能無効化 (xxxx.c 10行目: xxxxx関数)		
	委小		電圧が製品の規正値を超えた場合
	理由		製品仕様として15V以上がxx秒以上
説明		電圧が1.5V以上または1.2V以下	
電圧異常 (xxxx.c 20行目: xxxxx関数)	仕様理由説明		仕様理由説明
	電圧異常判定 (xxxx.c 50行目)		仕様理由説明
	過電圧異常検知		仕様理由説明
	低電圧異常検知		仕様理由説明
	正常時		仕様理由説明
	過電圧 (xxxx.c 100行目)		仕様理由説明
	ステータス遷移		仕様理由説明
	低電圧 (xxxx.c 200行目)		仕様理由説明
	ステータス遷移		仕様理由説明
	動作停止通知		仕様理由説明
電圧値監視	要求		電圧値はxxMS毎に最新の値を使用
理由			・製品の動作がxxMSのため
説明			・処理負荷低減のため、必要以上
取得電圧値チェック			仕様理由説明

仕様以下の文章は省略

有識者と要求、理由の正しさと必要性を確認していく

# 課題解決へのアプローチ ～機能の分類と必要性の整理

## 下記の内容について実施

- ・有識者と作成したUSDMの要求、理由が正しいかについて確認
- ・有識者と必要、不要、共通 機能の確認
- ・確認結果から変更点をまとめる
- ・構造の整理を実施

## 確認中の分類基準

必要な要求

不要な要求

変更が必要な要求

共通要求

整理中

異常判定機能	要求	0002_0003_0000	各種セン
	理由		
	説明		各種センサー値が異常値となっている
	機能無効化 (xxxx.c 10行目: xxxxx関数)		xxx仕様の場合は電圧異常検知関連 xxx仕様の場合は外部のICが管理す
	機能無効化時		xxx仕様の特別仕様であるため不要 説明
	要求		電圧が製品の規定値を超えた場合
	理由		製品仕様として15V以上がxx秒以上
	説明		電圧が1.5V以上または1.2V以下
	電圧異常判定 (xxxx.c 50行目)		仕様 理由 説明
	過電圧異常検知		仕様 理由
低電圧異常検知			
正常時			
異常判定は別機能統合なので ここでは確認して遷移機能とする			異常判定要求に統合 ⇒ここでは呼び出しだけ 【電圧異常判定】
電圧異常 (xxxx.c 20行目: xxxxx関数)			
↓			
電圧状態遷移			
過電圧 (xxxx.c 100行目)			仕様 理由 説明
ステータス遷移			仕様 理由
動作停止通知			
低電圧 (xxxx.c 200行目)			理由 説明
ステータス遷移			仕様 理由 説明
動作停止通知			仕様 理由
電圧値監視	要求		電圧値はxxMS毎に最新の値を使用
理由			・製品の動作がxxMSのため
説明			・処理負荷低減のため、必要以上
取得電圧値チェック			仕様 理由 説明

確認結果から変更点をまとめる

# 課題解決へのアプローチ ～機能の分類と必要性の整理

## 下記の内容について実施

- ・有識者と作成したUSDMの要求、理由が正しいかについて確認
- ・有識者と必要、不要、共通 機能の確認
- ・確認結果から変更点をまとめる
- ・構造の整理を実施

## 確認中の分類基準

必要な要求

不要な要求

変更が必要な要求

共通要求

		0002_0003_0000	各種セ	整理後
異常判定機能	要求			
	理由			
	説明			各種セの共通値が異常値となっている
	要求			電圧が製品の規定値を超えた場合
	理由			製品仕様として15V以上がxx秒以上
	説明			電圧が1.5V以上または1.2V以下
	仕様	電圧異常判定		
	理由			
	説明			
	仕様	状態変更		
理由				
説明				
仕様	ステータス遷移			
理由				
説明				
仕様	動作停止通知			
理由				
説明				
仕様	電圧値監視			
要求			電圧値はxxMS毎に最新の値を使用	
理由			・製品の動作がxxMSのため	
説明			・処理負荷低減のため、必要以上	
仕様	取得電圧値チェック			
理由				
説明				

構造の整理を実施

# 課題解決へのアプローチ ～機能の分類と必要性の整理

## 機能整理に大事だと感じたポイント

例：電圧異常判定の特殊仕様を削除、異常判定を統合、状態遷移をまとめた整理を実施した場合

### 整理前のUSDM(ソースコード情報付き)

異常判定機能	要求	0002_0003_0000	各種センサで異常値が計測された理由 説明
	機能無効化 (xxxx.c 10行目: xxxxxx関数)		各種センサ値が異常値となっていない理由 説明
	機能無効化時		仕様理由 説明
	電圧異常判定 (xxxx.c 50行目)		電圧が製品の規定値を超えた場合製品仕様として15V以上がxx秒以上電圧が1.5V以上または1.2V以下理由 説明
	過電圧異常検知		仕様理由 説明
	低電圧異常検知		仕様理由 説明
	正常時		仕様理由 説明
	過電圧 (xxxx.c 100行目)		仕様理由 説明
	ステータス遷移		仕様理由 説明
	動作停止通知		仕様理由 説明
電圧異常 (xxxx.c 20行目: xxxxxx関数)	過電圧 (xxxx.c 100行目)		仕様理由 説明
	ステータス遷移		仕様理由 説明
	動作停止通知		仕様理由 説明
	低電圧 (xxxx.c 200行目)		仕様理由 説明
	ステータス遷移		仕様理由 説明
動作停止通知		仕様理由 説明	

### 変更点資料

異常判定機能	要求	0002_0003_0000	各種センサで異常値が計測された理由 説明
	機能無効化 (xxxx.c 10行目: xxxxxx関数)		各種センサ値が異常値となっていない理由 説明
	機能無効化時		仕様理由 説明
	電圧異常判定 (xxxx.c 50行目)		電圧が製品の規定値を超えた場合製品仕様として15V以上がxx秒以上電圧が1.5V以上または1.2V以下理由 説明
	過電圧異常検知		仕様理由 説明
	低電圧異常検知		仕様理由 説明
	正常時		仕様理由 説明
	電圧異常 (xxxx.c 20行目: xxxxxx関数) ↓ 電圧状態遷移		電圧が製品の規定値を超えた場合製品仕様として15V以上がxx秒以上電圧が1.5V以上または1.2V以下理由 説明
	過電圧 (xxxx.c 100行目)		仕様理由 説明
	ステータス遷移		仕様理由 説明
動作停止通知		仕様理由 説明	
低電圧 (xxxx.c 200行目)		仕様理由 説明	
ステータス遷移		仕様理由 説明	
動作停止通知		仕様理由 説明	

### 整理後のUSDM

異常判定機能	要求	0002_0003_0000	各種センサで異常値が計測された理由 説明
	電圧異常判定		各種センサ値が異常値となっていない理由 説明
	電圧状態遷移		電圧が製品の規定値を超えた場合製品仕様として15V以上がxx秒以上電圧が1.5V以上または1.2V以下理由 説明
	状態変更		仕様理由 説明
	ステータス遷移		仕様理由 説明
	動作停止通知		仕様理由 説明
	過電圧 (xxxx.c 100行目)		仕様理由 説明
	ステータス遷移		仕様理由 説明
	動作停止通知		仕様理由 説明
	低電圧 (xxxx.c 200行目)		仕様理由 説明
ステータス遷移		仕様理由 説明	
動作停止通知		仕様理由 説明	



異常判定は別機能統合なのでここでは確認して遷移機能とする

xxx仕様の特別仕様であるため不要

異常判定要求に統合  
⇒ここでは呼び出しだけ【電圧異常判定】

状態に応じた処理を行う処理【状態変更】として統合

- ・ソースコードと整理後USDMでトレースできる資料を作成しておく
- この後整理した構成でモデルを作成する際に元となるコードの位置が分からなくなるため

# 課題解決へのアプローチ

設計書、ソースコード間のトレーサビリティ情報の欠如により仕様理解が難解

ソースコードを読み込まないと話しに参加できない

現状のプロジェクトメンバーの中に対象製品開発に対する知識、経験がある人がいない

ソースコード開発からSimulinkモデル開発への移行経験がない

## 解決プロセスの流れ

ソースコード  
複雑な機能が混在した状態の既存コード

1

USDMを活用したスペックアウト

2

機能の分類と必要性の整理

3

階層構造を活かしたモデル設計



# 課題解決へのアプローチ ～階層構造を活かしたモデル設計

## USDМからSimulinkモデルへの変換

USDМの階層構造を機能的なアーキテクチャにすることで、Simulinkモデルの階層構造と親和性が高くなり、自然な形で実装する事ができます。

## Simulinkモデル開発への移行メリット

- ・ブロック線図による直感的な可読性
- ・補足情報を自由に配置できる事で保守性の向上

## USDМとSimulinkモデル構造の対応イメージ

上位要求

機能レイヤ

下位要求

サブ機能レイヤ

仕様

データフローレイヤ

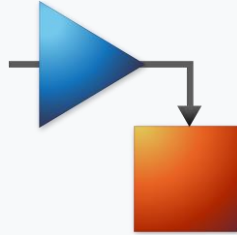
作成したUSDМをほぼそのままの構造でモデルへ実装が可能

# 課題解決へのアプローチ ～階層構造を活かしたモデル設計

## Simulinkとは

ブロックダイアグラムによるモデリング環境

- ・ モデルベースデザインのプラットフォーム
- ・ ブロック線図でシステムを視覚的に設計
- ・ 時間軸上でのシミュレーションが可能
- ・ 複雑なシステムも階層的に表現可能



## 業界での活用分野



自動車工学



ロボティクス



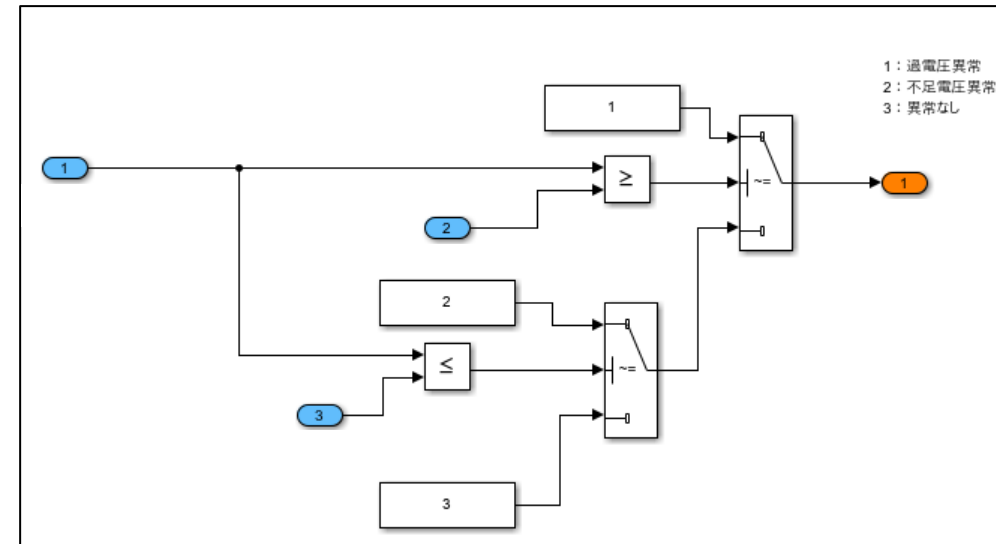
電力システム



産業オートメーション

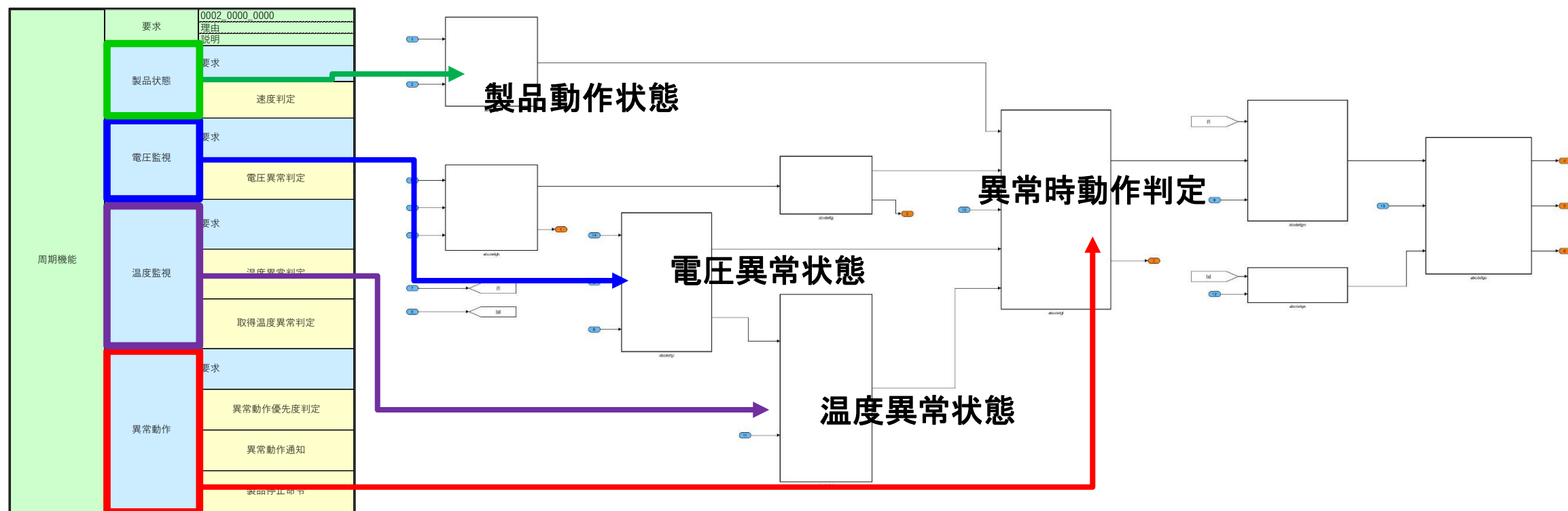
## 従来手法との比較

	手動コーディング	ブロックモデリング
直観的理解	△ 専門知識が必要	◎ 視覚的に把握可能
修正の容易さ	△ コード全体の理解が必要	○ ブロック単位で修正可能
検証効率	× 実装後の検証が中心	◎ 設計段階から検証可能
保守性	△ ドキュメント依存	○ モデル自体が文書化



AIにSimulinkモデルの説明をしてもらった結果

# 課題解決へのアプローチ ～階層構造を活かしたモデル設計

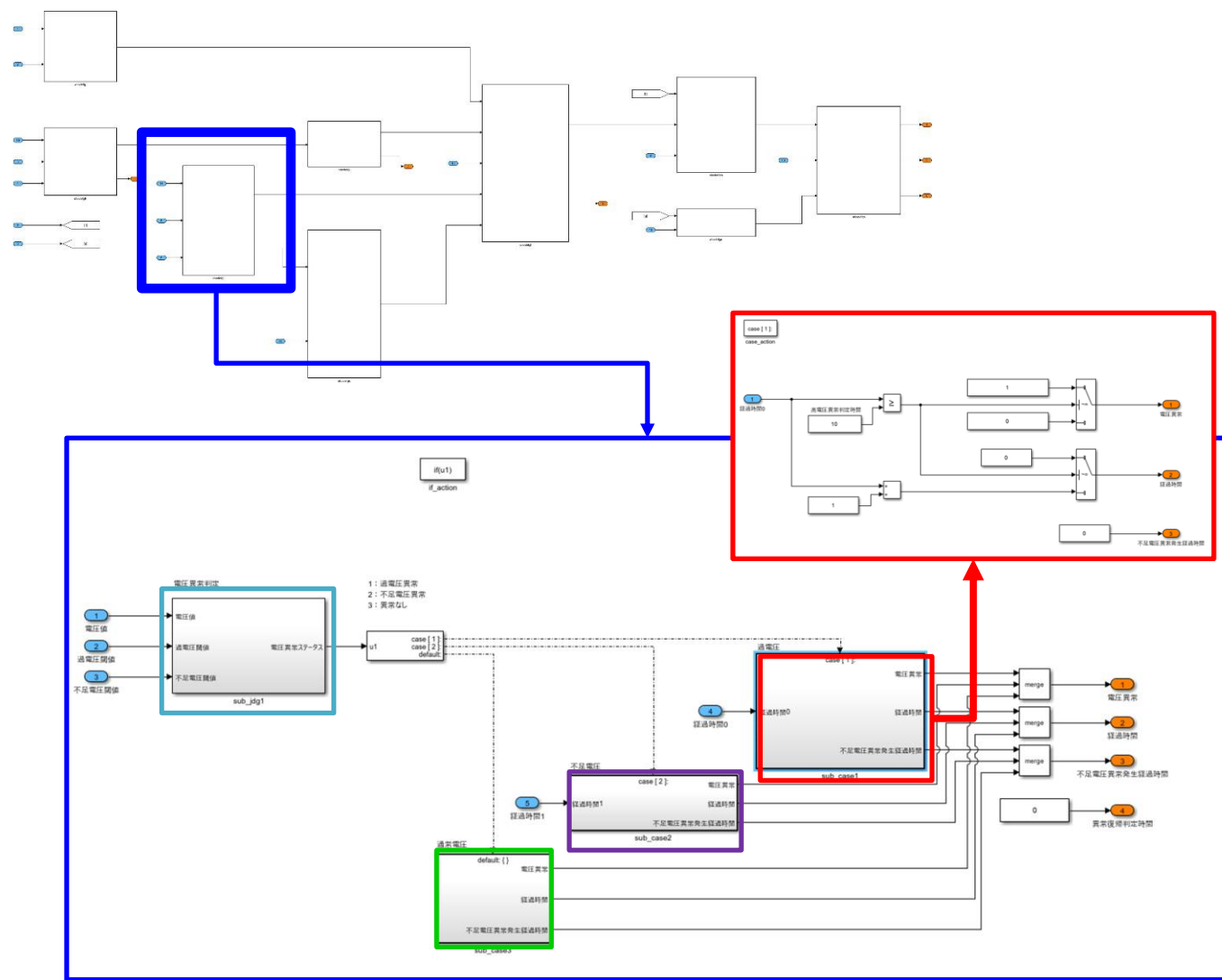


- ・USDM作成時の機能をまとめる際に並行して簡易的にデータの流れをモデルで作成しておく  
情報を追加していく事でDFDの代わりにもなるので便利

作成したUSDMを元に情報の流れをモデルで作成

# 課題解決へのアプローチ ～階層構造を活かしたモデル設計

電圧監視	要求	0002_0003_0000	製品の最短周期毎に電圧を監視す
		理由	製品仕様として15V以上がxx秒以上
	説明	電圧値の取得、取得した電圧値の	
	要求	電圧が製品の規定値を超えた場合	
	理由	製品仕様として15V以上がxx秒以上	
	説明	電圧が1.5V以上または1.2V以下	
	電圧異常判定	電圧状態確認	仕様 理由 説明
		過電圧	仕様 理由 説明
		異常電圧時間更新	仕様 理由 説明
		過電圧異常状態遷移	仕様 理由 説明
		低電圧	仕様 理由 説明
		異常電圧時間更新	仕様 理由 説明
		低電圧異常状態遷移	仕様 理由 説明
		通常電圧	仕様 理由 説明
異常電圧時間更新		仕様 理由 説明	
通常電圧状態遷移		仕様 理由 説明	



下位要求、仕様を実装

# 課題解決へのアプローチ ～機能の分類と必要性の整理

USDMからソースコードの検索をしている例

ソースコード

```

VoltageStatus check_over_voltage(uint16_t raw_voltage) {
    /* 電圧値の有効範囲チェック */
    if (raw_voltage > 4095) {
        return VOLTAGE_OVER; // 異常値検出
    }

    /* 実際の電圧値に変換 (例: 5V基準の場合) */
    uint32_t voltage_mv = (raw_voltage * 5000UL) / 4095;
    /* 過電圧判定 */
    if (voltage_mv > OVER_VOLTAGE_THRESHOLD) {
        return VOLTAGE_OVER;
    }

    return VOLTAGE_NORMAL;
}
    
```

変更点資料

要求	0002_0003_0000	各種センサで異常値が計測された
理由		
説明		
機能無効化 (xxxx.c 10行目: xxxxx関数)		各種センサ値が異常値となってい xxx仕様の場合は異常検知関連 xxx仕様の場合は外部のICが管理す
機能無効化時		xxx仕様の特別仕様であるため不要
要求		電圧が製品の規定値を超えた場合
理由		製品仕様として15V以上がxx秒以上
説明		電圧が1.5V以上または1.2V以下
電圧異常判定 (xxxx.c 50行目)		仕様 理由 説明
過電圧異常検知		理由
低電圧異常検知		理由
正常時		説明
電圧異常 (xxxx.c 20行目: xxxxx関数)		仕様 理由 説明
電圧状態遷移		仕様 理由 説明
過電圧 (xxxx.c 100行目)		仕様 理由 説明
ステータス遷移		仕様 理由 説明
動作停止通知		仕様 理由 説明
低電圧 (xxxx.c 200行目)		理由 説明
ステータス遷移		仕様 理由 説明
動作停止通知		仕様 理由 説明

整理後のUSDM

要求	0002_0003_0000	各種センサで異常値が計測された
理由		
説明		
要求		各種センサ値が異常値となってい
理由		電圧が製品の規定値を超えた場合
説明		製品仕様として15V以上がxx秒以上
電圧異常判定		電圧が1.5V以上または1.2V以下
仕様		
理由		
説明		
電圧状態遷移		仕様 理由 説明
状態変更		仕様 理由 説明
ステータス遷移		仕様 理由 説明
動作停止通知		仕様 理由 説明



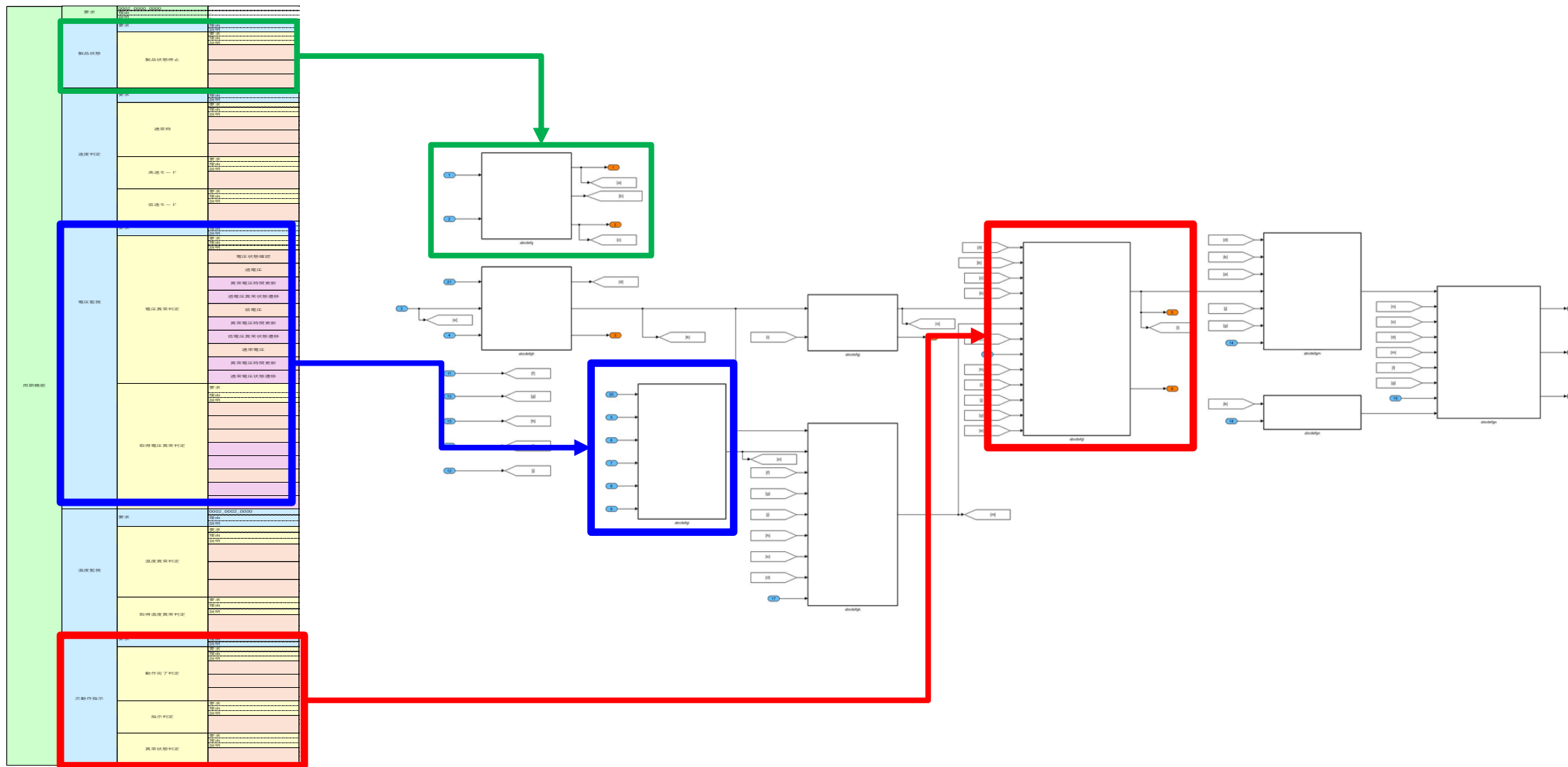
異常判定要求に統合  
⇒ここでは呼び出しだけ  
【電圧異常判定】

状態に応じた処理を行う処理  
【状態変更】として統合

異常判定は別機能統合なので  
ここでは確認して遷移機能とする

仕様については変更点資料からソースコードを検索して実装

# 課題解決へのアプローチ ～階層構造を活かしたモデル設計



下位要求、仕様を実装した事で確定した入出力を結線

# 課題に対しての結果

設計書、ソースコード間のトレーサビリティ情報の欠如により仕様理解が難解

ソースコードを読み込まないと話しに参加できない

現状のプロジェクトメンバーの中に対象製品開発に対する知識、経験がある人がいない

ソースコード開発からSimulinkモデル開発への移行経験がない

## 解決プロセスの流れ

ソースコード  
複雑な機能が混在した状態の既存コード

1

USDMを活用したスペックアウト

2

機能の分類と必要性の整理

3

階層構造を活かしたモデル設計

# 課題に対しての結果

コード差分開発からの脱却するきっかけが出来た

関係者間のコミュニケーションが向上した

有識者が担当者として不在のプロジェクトで機能再設計が行えた

ソースコードからモデルへの移行プロセスを作成できた

## 解決プロセスの流れ

ソースコード  
複雑な機能が混在した状態の既存コード

1

USDMを活用したスペックアウト

2

機能の分類と必要性の整理

3

階層構造を活かしたモデル設計



# 上手くいかなかったこと

---

1. スペックアウト時に要求を読み取ることができなかった結果、構造が複雑なモデルになってしまった。
2. 物を直接動作させる機能について知識不足により、うまく再整理できなかった。

# 上手くいかなかったこと(要因)

- **要求のグループ化不足**
  - 要求と仕様の区別が不明確
  - モデル構造の複雑化
  - 小粒度の要求が多数発生
- **技術知識の不足**
  - ハードウェア連携部分（ライト、モーター、センサ等）の理解不足
  - 既存コードをそのままUSDM化した箇所が存在
- **作業者間の認識差**
  - 要求・仕様の粒度の不一致
  - XDDP/USDMへの関心度による取り組み姿勢の違い
  - 必要/不要の判断基準のばらつき
- **適用範囲の見極め**
  - アプリケーション層：適用可能性が高い
  - ミドル層以下：適用難易度が高い
  - 他製品への横展開可能性の調査

# まとめ

---

今回の経験を通してお伝えしたいことは2点です。

## まとめ

# USDMを活用して開発が楽しく

仕様変更が明確になり、本質的な検討が可能、レビューが活性化

目的意識が重要

仕様理解と機能再整理を目的としたUSDM化は楽しい

単なるソースコードのUSDM化は効果が薄い

## 時間不足

派生開発の複雑さと時間不足の現実

成功の条件

ソースコードの熟知

影響範囲がわかる資料→XDDPで定義された成果物作成の時間確保

# 参考資料・サービス

## AFFORDD研究会資料

affordd-t2-usdmtext-basic\_1.3.pdf

affordd-t2-usdmtext-appendix\_1.3.pdf

t20booklet\_xddp\_guide\_ver10.pdf

## EurekaBox

<https://www.eureka-box.com/>



## USDM 小冊子

### 基礎編



ver 1.3



むすび丸

笑顔咲くたび  
伊達な旅

仙台・宮城

Sendai & Miyagi, where smiles blossom

# End of Presentation

最も強い者が生き残るのではなく、最も賢い者が生き延びるのでもない。  
唯一生き残ることが出来るのは、変化できる者である。

## RSビジョン

3つのRS Creatingにより真の情報サービス企業となる

－「データ処理」から「情報創造」へ－

