

## <セッション>

### 制御モデルの仕様化と派生開発への展開

#### ～Advanced Drive コントローラ開発への USDM、XDDP の適用～

##### <質問1>

”要求獲得の混乱”とは具体的にどういうものですか？複数のステークホルダーからでてくる要求間の不整合とかですか？

##### <回答>

仰る通りの事象や、要求発行回数が増えることで過去と現在の要求間の干渉、矛盾が発生することです。

##### <質問2>

制御モデル特有の課題がある中で、モデルベース開発採用された理由・モチベーションは何でしょうか？

##### <回答>

仕様設計と検証の高速化がモチベーションでした。

また、プログラミング言語として平易であること、制御設計用のライブラリが用意されていて実装が楽ということも理由でした。

##### <質問3>

制御モデルから USDM を作成する場合、その要求の意図、理由は取れるのでしょうか。モデルだけでは、その背景理由が失われていることが多いように思います

##### <回答>

仰る通りモデルだけでは読み取れないので、設計書やモデルに書かれたコメントも参考にして意図、理由を引き出しました。それでも分からない場合は、設計者やシステム要求部署にヒアリングして引き出しました。

##### <質問4>

モデルベース以前はどんな手法で開発をしていましたか？

モデルベースだと目的や意図が伝わりづらいと言っていたので以前の手法はそこが強みだと思ったのですが、以前の特徴等を教えて下さい。

##### <回答>

従来のシャシー制御の開発では、ステークホルダーが少なかった事もあり、自分たちがシステム主管部署で要求仕様書を発行しサプライヤさんに実装いただく、という開発プロセスでした。要求が明確だったというのが強みだったと思います。

#### <質問5>

USDM 作成時において、複数のグループを横断するような要求や実装仕様は存在しましたでしょうか？存在していた場合、その表現はどのように工夫されましたか？

#### <回答>

存在しました。例えば、あるモードに応じて複数の機能グループのふるまいを変更するというような要求です。この場合、論理モデル上でふるまいを変える部分を見る化（色付け）し、見える化できた論理モデルのふるまいをどう変えるかを USDM で記載する、のような工夫をしました。

#### <質問6>

USDM を自然言語で記載することで、担当者間の認識間違いや誤解釈は発生しなかったでしょうか？

#### <回答>

共通認識にするために、自然言語の表現は難しかったです。

一人だけで作成すると、客観性が乏しい為、複数メンバーでレビューしながら、共通認識にしていきました。

また、古畑さんにもアドバイスをいただきながら進めました。

記載レベルを揃えるために、分割基準を決め、記載のパターンをテンプレート化して表現を統一することもしました。

#### <質問7>

今回は変更要求仕様書と TM をモデルからボトムアップで作成することでトレーサバリティがとれた要求仕様とモデルの管理ができるようになったと思います。今後はトップダウンで要求から仕様を検討し、TM で関係するモデルを修正していくという流れを想定していますか？そのときに課題になりそうなことはどんなことが考えられるでしょうか？

#### <回答>

今回、要求仕様書の作成は、ボトムアップでしたが、変更要求仕様書は、トップダウンで実施しました。現状、変更要求仕様書作成をトップダウンで継続実施していますが、課題になりそうなものは今のところ見つかっていません。

仮に、今回と同じパターンの開発の場合どうするかは、その時の状況（納期、ソフトの完成度、ドキュメントの完成度、設計者のレベル etc ）に応じて判断することになると思います。

#### <質問8>

XDDP により、レビューの精度が向上したとのことですが、自己検出力向上という視点で何か気づきはなかったでしょうか？

#### <回答>

影響範囲を TM を使って把握できるため、自己検出においても効果を感じました。

理由、説明などの欄があり、上位要求とのつながりを考え、システム目線での記載をすることでセルフレビューしやすくなる、という気づきもありました。

<質問 9>

動的設計は SysML など別の記法を利用されていると思うのですが、それと USDМ はどのように連携されていますか？

<回答>

状態遷移図（ステートマシン図）は SySML と同じものですし、論理モデルは、SysML の内部ブロック図に近い形です。今回は、USDМ の構成と SysML を対応させるような使い方をしたととらえています。

<質問 10>

USDМ で表現しにくい要求がありませんか？

<回答>

状態遷移など動的なふるまいの表現は、難しかったです。

<質問 11>

論理モデルも、変更されていくと思いますが、それは、どのタイミングで行われるのでしょうか？ぎりぎりまで、論理モデルは変えないと思うのですが、その変化点のレビューとかはどのようにされていますか

<回答>

変更要求仕様書を作成し、実装、検証完了後、機能仕様書に変更要求仕様を反映するタイミングで、論理モデルも修正しました。修正はこのタイミングですが、レビューについては、変更要求仕様書のレビューの中で実施していました。

<質問 12>

実装仕様から論理モデル（論理設計）を作ることがポイントと思いました。現実的なアプローチと思いますが、論理設計を中心にすすめるアプローチにも取り組んでいるのでしょうか？

<回答>

XDDP に移行してからは、論理設計を中心にすすめるアプローチになっていると考えています。

<質問 13>

不具合ゼロという素晴らしい結果が得られていますが、工数の面ではどうだったでしょうか？（TM の作成など工数が増えた部分と、手戻り削減による工数削減した部分のバランスを知りたいです）

<回答>

工数の測定はできてはいませんので、定性的な回答になります。

論理モデルや USDМ、TM の作成の段階においては、工数が増えてはいますが、その後の XDDP で手戻りがないこと、不具合がないことで工数削減につながっていると考えています。全体としては、工数削減効果の方が大きいととらえています。

#### <質問 1 4>

浅野さんがおっしゃる「USDM+TM」・・・、これがベリザープの堀川さんがおっしゃった「システムの振る舞い」を形式知化したものに近いのかも？と理解しました。また、TM というものを、堀川さんがおっしゃった「要求内容を品詞レベルにまで分解」する行為に匹敵するように思います。こんな認識で合ってますかね？

#### <回答>

USDM という要求は「あるドメイン（今回の場合は CTL）のふるまい」を書くので、ご認識の通りです。「品詞レベルへの分解」との関係性を考えてみますと、抽象化という意味で、実装モデルからグルーピングして作った論理モデルに近いのではないかと思います。今回の説明でお伝えした TM はあくまで、実装の場所と機能仕様を紐づけるマトリックスです。

#### <質問 1 5>

機能仕様＝制御仕様は、実現手段を規定していて、上位要求のひとつの実現手段の選択のように思います。特にコアロジックは様々な要求に紐づくと思います。それをどのように要求にまとめあげたのか、一例があれば教えてもらえると有難いです。

#### <回答>

コアロジックを決めるに当たって、要求の矛盾は起こり得て、そこは優先順位、実装都合などを鑑みて、要求を調整しました。

#### <質問 1 6>

XDDP からソフトウェアプロダクトライン開発への移行の方法論はあるでしょうか。（SPL のスペシャリストの方が質問者の中におられるようにお見受けしましたが。。）

#### <回答>

漸次的移行プロセス XDDP4SPL という方法論が、現福岡大学の中西恒夫さんから提唱されています。インターネット検索してみてください。過去の派生開発カンファレンスでも一部発表されています。

[https://affordd.jp/forum/2013\\_04/forum2013\\_04\\_xddp4spl.pdf](https://affordd.jp/forum/2013_04/forum2013_04_xddp4spl.pdf)

方法論ではありませんが、「XDDP と SPLE の連携・移行・使い分けガイドの紹介」という発表を、派生開発推進協議会 T14 研究会」が発表しており、ご参考となるかもしれません。

[https://www.affordd.jp/conference2014/affordd\\_conference2014\\_p2.pdf](https://www.affordd.jp/conference2014/affordd_conference2014_p2.pdf)

#### <質問 1 7>

USDM の逆おこしというのがイマイチ理解できませんでした。要件を逆から起こすとは設計からでしょうか。それは要件ではなく、機能設計・詳細設計ではないでしょうか

#### <回答>

設計されたモデルから実装仕様を抽出し、それをグルーピング、抽象化することで要求を引き出しました。ですので、逆起こしと申し上げました。とはいえ、モデルだけから要求を引き出すのにも限界があるので、設計書も参考にしていました。