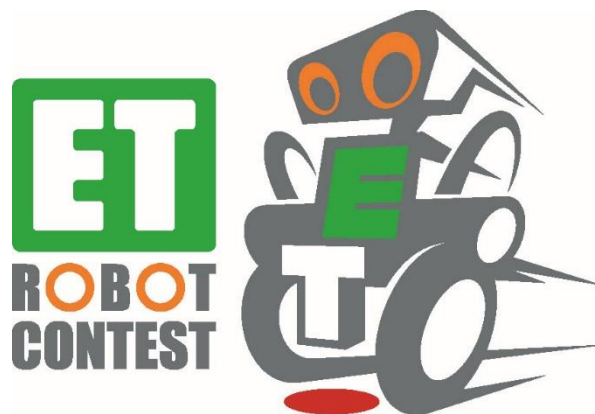




ETロボコン2021 モデル審査

ETロボコン実行委員会

背景：求められている人材・解決策

- 近年、技術変化などにより事業環境は変化し続けている
- そのような中で現在・将来ともに不足している人材は
「システム全体を俯瞰して思考できる人材」との結果
プロジェクトマネージャ・設計技術者も不足
- また、開発の課題解決に「既存スキルの向上」「プロジェクトマネージャのスキル向上」「開発手法・開発技術の向上」が求められている
- こうした人材の育成・スキル向上が急務

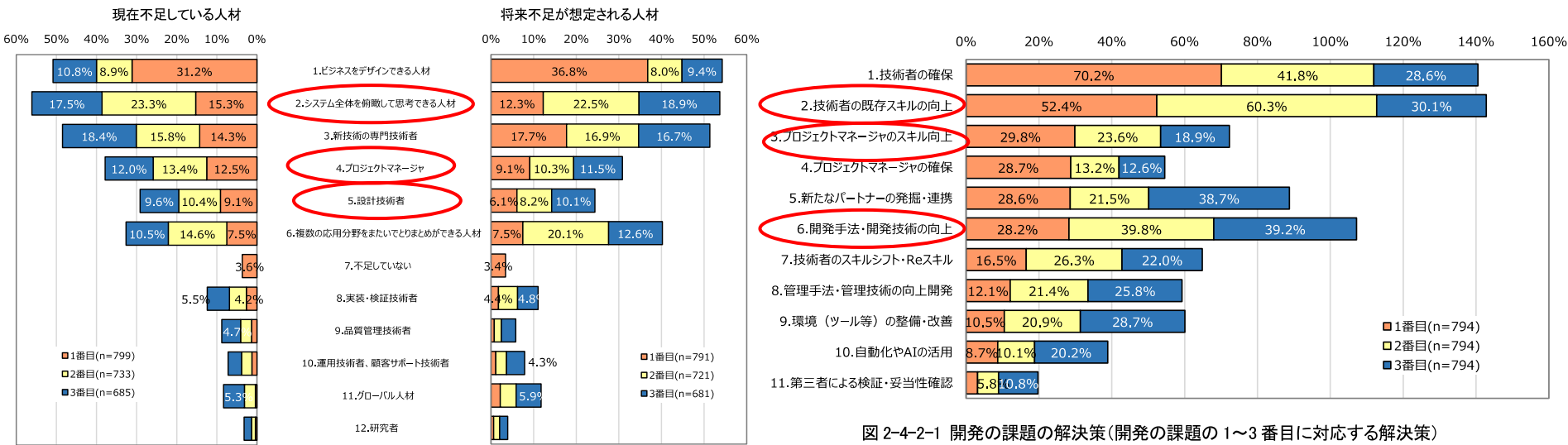


図 2-6-3-1 現在不足している人材/将来不足が想定される人材(全体)

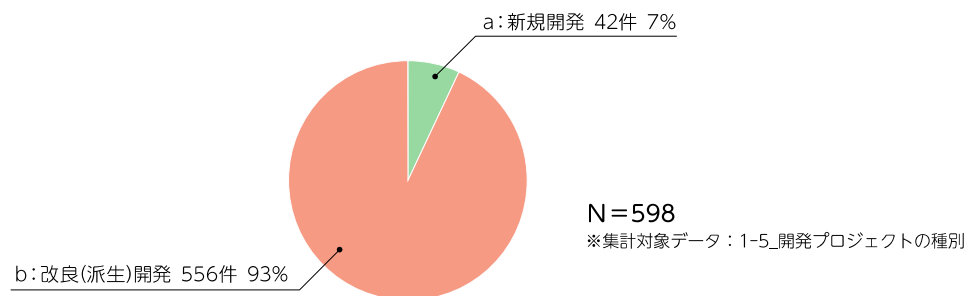
図 2-4-2-1 開発の課題の解決策(開発の課題の1～3番目に対応する解決策)

※IPA「2019 組込み/IoTに関する動向調査 調査報告書」(https://www.ipa.go.jp/ikc/reports/20200331_2.html)より引用

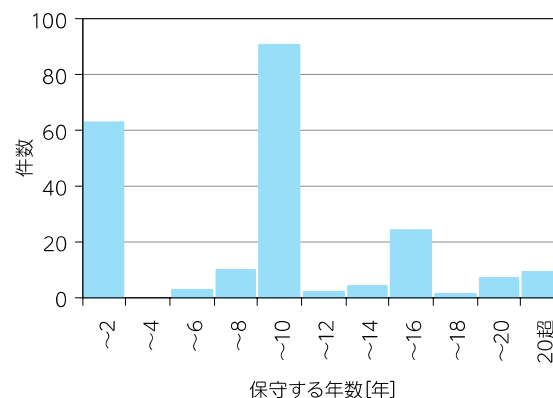
背景：開発現場の状況

- 近年の組込み機器の高度化に伴い、ソフトウェアは大規模化・複雑化が進んでいる
- 開発プロジェクトも既存のソフトウェアを改良する「派生開発」が大部分を占める
- 保守年数も10年程度になるものが多い
→ 特に若手がシステム全体を構築する機会が少なく、各種スキルを学んでも業務での活用ができないためモチベーションが上がらない

図表4.2-1 開発プロジェクトの種別



図表4.2-3 開発システムの保守年数

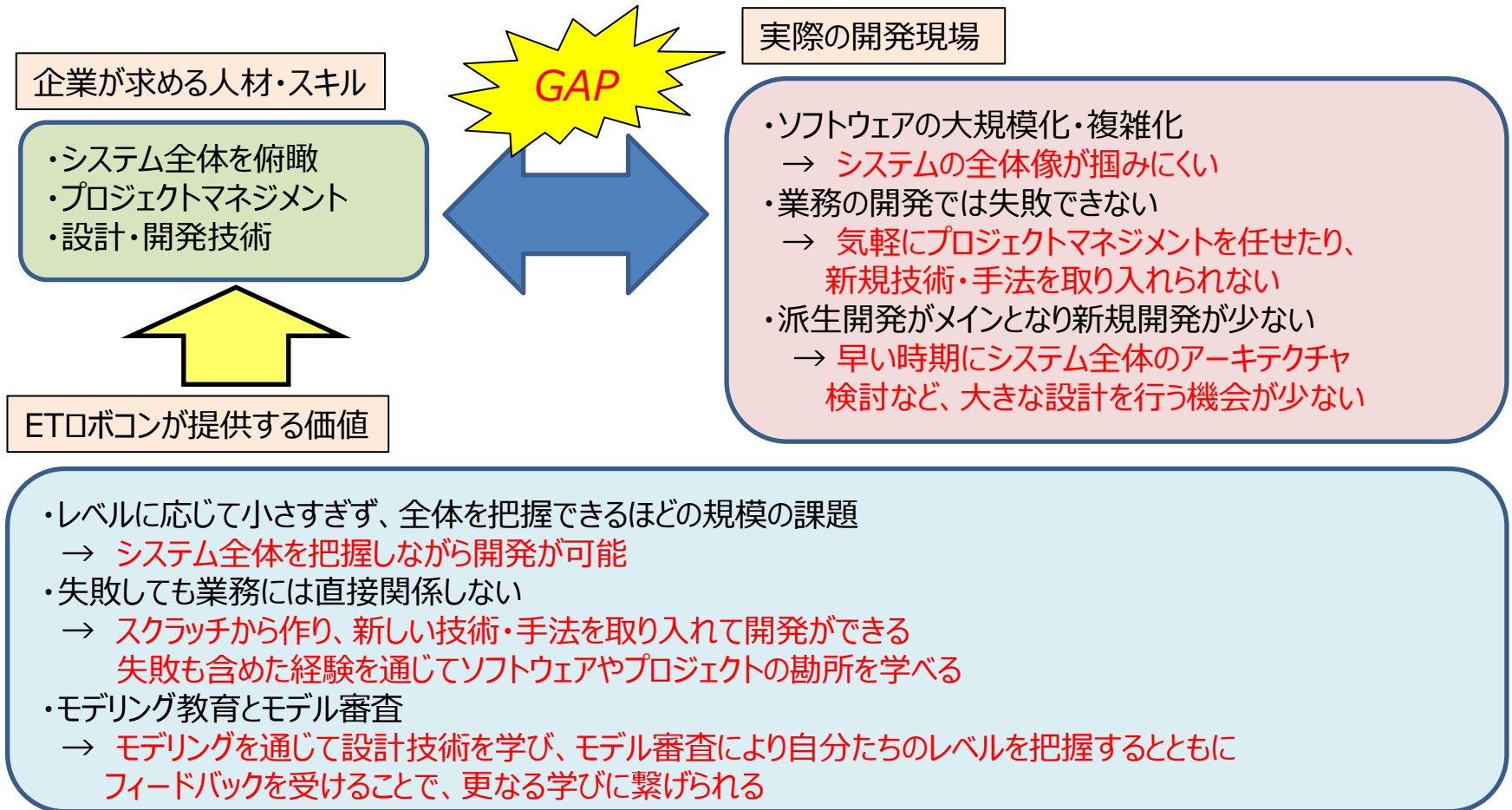


N=214
※集計対象データ：1-5b_保守する年数

※IPA「組込みソフトウェア開発データ白書2019」(<https://www.ipa.go.jp/ikc/reports/20191119.html>)より引用

ETロボコンが提供する価値

- 企業が求めている人材像・スキルに対して、業務での育成が難しくなっている
- ETロボコンでは競技課題を題材にプロジェクト経験を積むとともに、**モデリングを通じた設計・開発技術向上の機会を提供する**



スキルに応じた 3 クラス

- 獲得したいスキルに応じた 3 つのクラスを提供
- 連続して参加して上のクラスに参加しても良いですし、同じクラスで前年の反省を踏まえた取り組みにしても良いでしょう

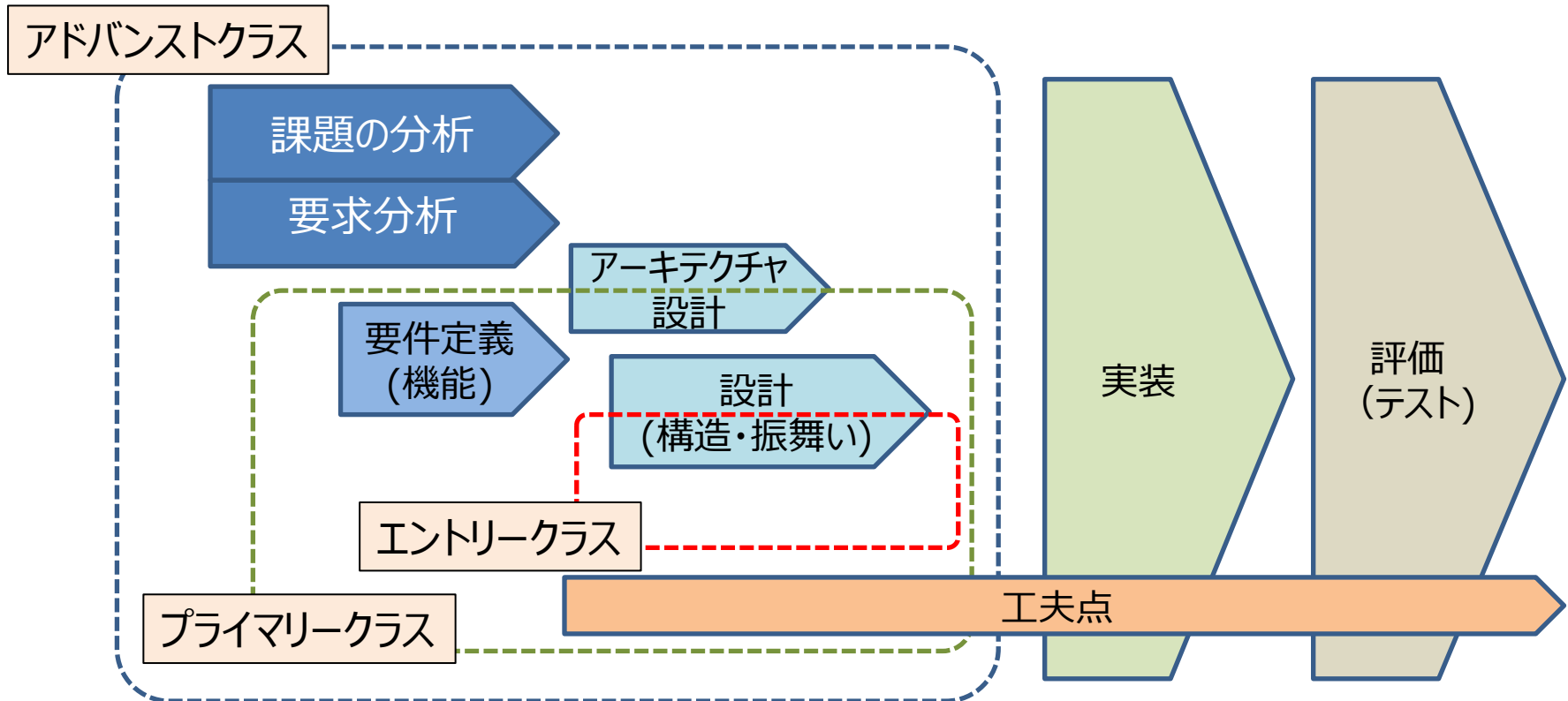
(ただし、エントリークラスは原則ETロボコン未経験者を対象としています)

クラス名	狙い	想定対象者	モデリングに要する期間 (参考)※
エントリー	・モデルを読んで理解し、改良することでモデルの使い方を体験する ・プログラムを作り動かす体験をする ・設計と実装の繋がりを把握する	モデリング未経験者 ソフトウェア開発初心者	1～4週間
プライマリー	・競技課題に合わせてモデルを構築する事で、モデリングの基礎を身につける ・品質向上に向けた取り組み	モデリング初級者	4週間～
アドバンスト	・モデルを使って課題解決ができる ・要求に基づき設計方針を立て、それに基づいた設計を実践できる	モデリング中級者	6 週間～

※エントリーはA評価、プライマリー/アドバンストはB+評価以上を取ったチームのアンケートから算出しています
現状のスキル・利用できる時間・環境・過去参加状況などによって増減しますので、一つの目安と考えて下さい
これ以外に環境構築（1ヶ月程度）、競技用プログラム作成にはモデリングと同程度の時間が必要です

各クラスのモデル審査対象範囲

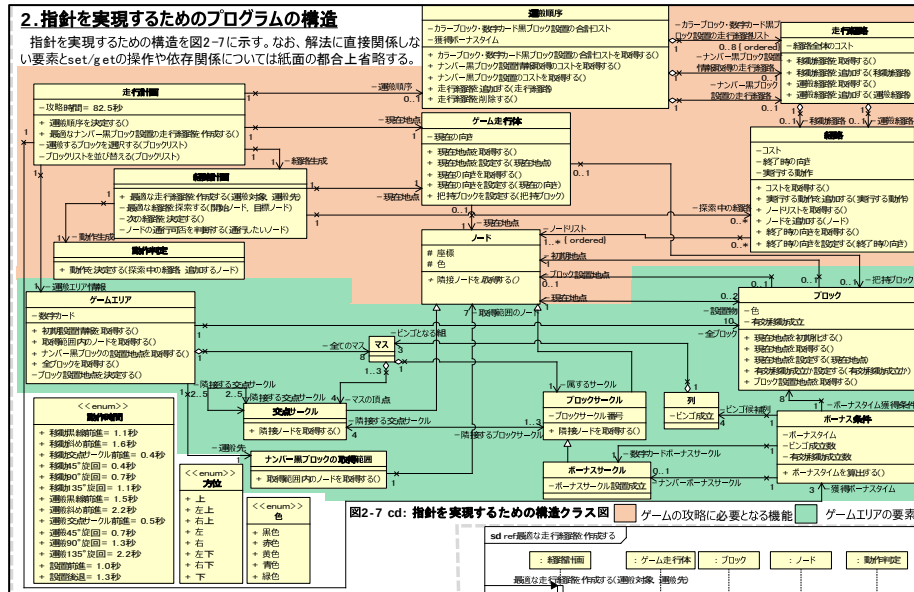
- クラスによってモデル審査対象は異なり、アドバンストクラスが最も範囲が広がります
- モデル審査では実装は審査対象外ですが、技術教育ではモデルと実装のつながりを教育します



※この図は対象範囲をイメージ化したもので、開発プロセスそのものではありません

要求モデル

- ## 分析モデル



課題のモデリングを通じて、対象を正しく理解し
シンプルで品質の高いアーキテクチャ・設計に繋げ
ます

2020作成モデル例（工夫点）

- アドバンストクラス・プライマリークラスでは性能向上などの取り組みである「工夫点」も評価します（プライマリークラスはオプションです）
- 設計という観点よりも、問題に対して論理的にアプローチしているかを評価します

4-1. 機械学習によるパラメータの自動調整

bl_13.3 自動パラメータ調整

走行タイム27秒を効率よく達成するために機械学習によるパラメータの自動調整・最適化を行った。

要件	方針	工夫点
・走行の調整をしやすい ・より速くするように調整する	効率的パラメータ調整 で速い走行を実現する	・パラメータ調整効率化 [4-1-2] ・パラメータの最適化 [4-1-3]

4-1-1. 課題と対策

従来・走行体の制御にはPID制御が用いられる
・各パラメータ値（走行体の速度、 K_p , K_i , K_d etc.）を主に人手によって経験則に基づきトライ＆エラーで調整
・・・**限界感度調整法** など
 K_p の値を調整し、そこから経験則に基づいて求められた関係式から K_i , K_d の値を求める手法。

限界感度調整法の特徴
○ K_i , K_d の調整はほぼ不要。時間はさほどからない
→ 20分もかからない程度

△ K_p の調整は、トライ＆エラーで何度も微調整する必要があり、人間がつきっきりになりカン・コツに依存。

△ パラメータの値の組合せが限定される
→ 他にもっと良い解があるかもしれない

課題

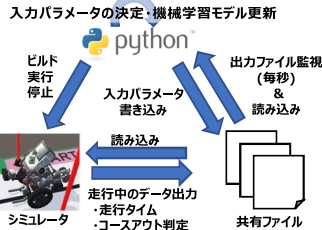
対策
・自動で繰り返しシミュレーションできる環境の構築
・機械学習によるパラメータの最適化

- 全てのプロセスを完全自動化（寝てでもできる）
- 機械学習の利用（ヒトのカン・コツに基づかない）
- 連続的な空間から最適なパラメータを探索（自由な値の組合せ）
- 様々なパラメータ調整にも適用可能（PIDに限らない）

4-1-2. シミュレーションの自動化環境の構築

シミュレータ環境の利点を最大限活用し、以下のサイクルを完全自動化

1. 入力パラメータの共有ファイルへの書き込み
2. シミュレーションのビルド & 実行
3. 走行中に出力されるデータを監視
→ コースアウト判定、ゴール判定 → シミュレーションの終了
4. シミュレーション結果で機械学習モデルを更新
5. 更新したモデルで次に入力すべきパラメータを算出
6. 1.に戻る



4-1-3. 機械学習によるパラメータの最適化

最適化アルゴリズムの選択

候補1. グリッドサーチ

事前に決めたパラメータの組合せを全探索する
△ データ候補点が離散的 △ 膨大な計算時間
例: K_p, K_i, K_d を10分割、30秒/回のシミュレーションで8h以上

候補2. ベイズ最適化

データを獲得しながらブラックボックス関数を逐次的に最適化する手法
○ データ候補点が連続的
○ 以下の戦略により効率的に最適化が可能

モデリング

ブラックボックス関数 f をガウス過程モデルでモデリング
→ x における f の値 $f(x)$ を正規分布に従う確率変数とみなし、 $f(x)$ の予測値とその予測の不確実さを表現可能なモデル

効率的な最適化のための戦略

次に出力 y を獲得する(=シミュレーションを実行する)ための入力 x の選び方:

$$x_{t+1} = \arg \max_x a(x)$$

$a(x)$: 獲得関数

“活用と探索の期待”を表現する評価指標

活用 $f(x)$ の値が期待的に小さい $\Rightarrow a(x)$ は大きくなる
= “最適解っぽい”ことがわかってきた領域を探索する

探索 $f(x)$ の分散が大きい $\Rightarrow a(x)$ は大きくなる
= 不確実性が高い (= 探索が不十分な) 領域を探索する

両者を加味することで効率よく最適解を探索

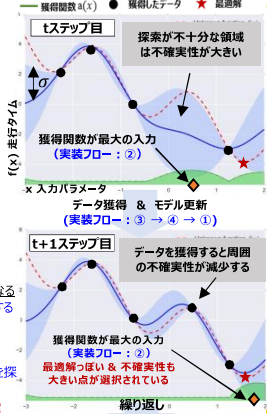
パラメータ最適化の方針

・パラメータ (K_p , K_i , K_d など) を入力し、シミュレーションを実行すると、走行タイムが出力されるという関係を、ブラックボックス関数 f として考える。
・この関数 f を最小化することで、走行タイムを最小とする入力パラメータを求める。



ベイズ最適化の流れ

真の関数 $f(x)$ 、観測したデータ D 、予測平均 $\mu(x)$ 、予測分散 $\sigma(x)$ 、最適解 x^*



実装フロー

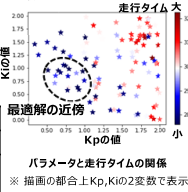
- ① 初期値としてランダムに選んだ n 個の入力 x とそれに対する出力 y を獲得し、学習データセットを作成
- ② 獲得したデータを使ってガウス過程モデルを更新
→ $f(x)$ の予測平均 $\mu(x)$ 、予測分散 $\sigma(x)$ 、獲得関数 $a(x)$ が求まる
- ③ $a(x)$ を最大化するような点を求め、その点を次の入力パラメータに決定
- ④ 決定したパラメータでシミュレーションを実行し、走行タイムを得る。コースアウトした場合は、ランダムに入力パラメータをサンプリングし、再度シミュレーションを実行
- ⑤ 入力パラメータと獲得した走行タイムの組を学習データセットに追加
- ⑥ ①～④を m 回繰り返し、走行タイムが最小なパラメータを出力

検証結果

設定
・ベイズ最適化と限界感度調整法で調整したパラメータ (K_p, K_i, K_d) における走行タイム、調整にかかった時間を比較
・ベイズ最適化は100個の点を獲得するまで繰り返し、最小タイムを出力

結果
・限界感度調整法と比べて時間がかかるが、より良いパラメータを探索できた
・右図から、空間を広く探索しながらも、最適解がありそうな領域を重点的に探索するという、ベイズ最適化の特徴も確認できた

	限界感度調整法	ベイズ最適化
走行タイム	27.699 秒	27.595 秒
計算時間	< 20 分 (手動)	3時間43分 (全自動)



課題・対策とその検証結果が論理的であるか、効果の有効性などを評価

技術教育

技術教育

2021年度の技術教育は全てリモートで行います（動画配信またはZoomを使った講義）

①全国共通(動画配信)

a. 走行体の基本制御とモデルによる可視化

b. モデル作成とそれをコードに変換する基礎演習

全国共通教育は動画配信で行います

内容は2020年作成したものがベースとなりますが、2021年も追加コンテンツを提供予定です

②地区独自(Zoomを利用)

地区ごとに独自の教育・フォロー会・モデル相談会を開催します（地区の状況によりできない場合もあります）

上記には、**初学者のためのセットアップサポートや演習**も含まれます。

提供資料

①環境構築ガイド・シミュレータ利用ガイド

②2020年度の提出モデルおよび評価付きコメント

③2020年CS大会ワークショップ資料

④走行体制御に関する参考資料

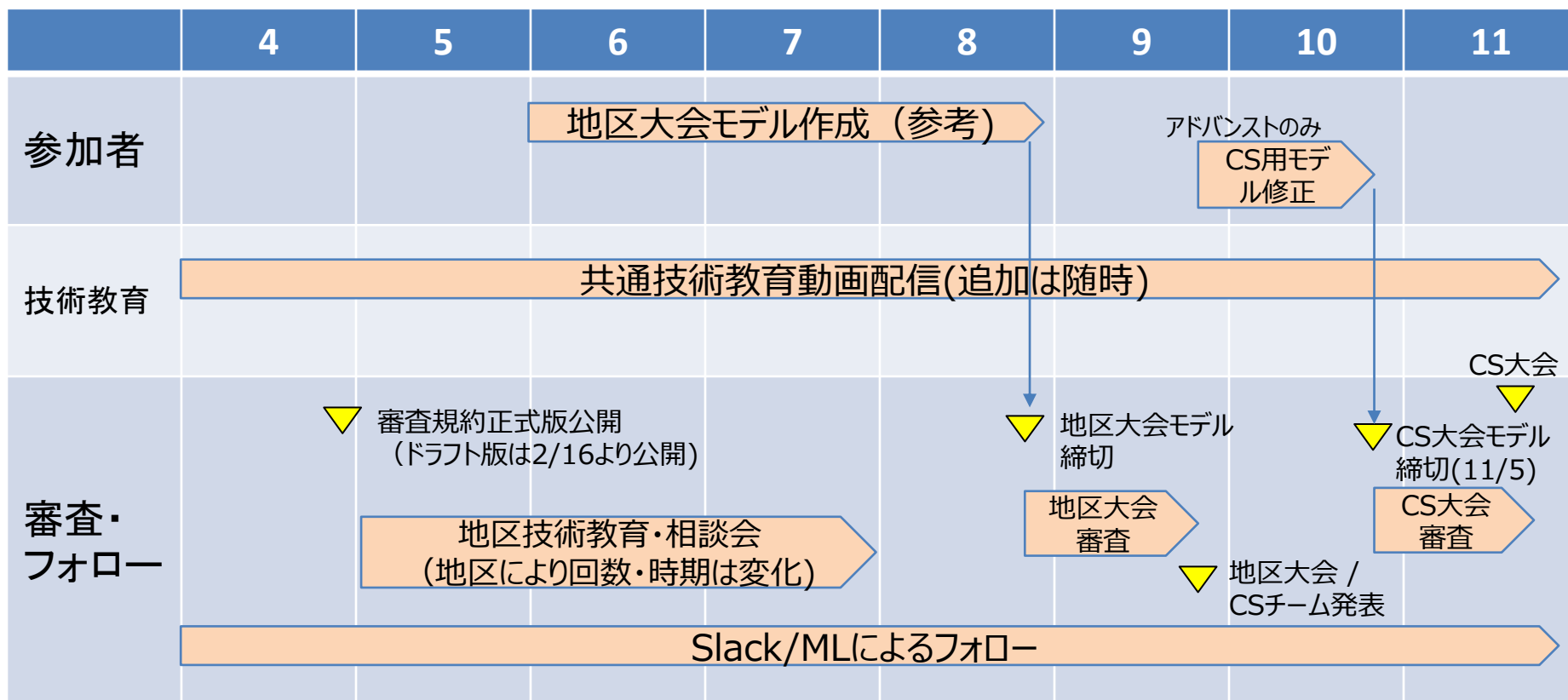
参加者サポート

①メーリングリストによる連絡事項展開・質問応答（チームから2名まで）

②Slackのワークスペースによる質問応答、情報共有（全参加者が登録可能）

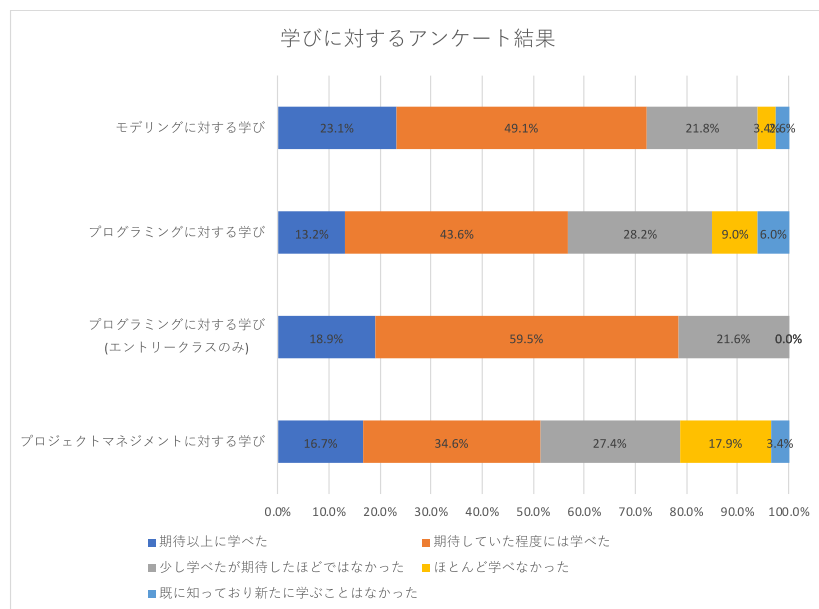
モデル教育・審査・フォローの大まかなスケジュール

- 2021年は4月の段階で審査規約・技術教育動画を展開
- モデルの学習や作成に数ヶ月の期間を確保



昨年のアンケートより

- 8割の参加者が各チームの参加目的を達成したと回答
→ 否定的な回答の多くは「時間がなく取り組めなかった」というものであったため、
計画的な取り組みが必要となります
- モデリングに関しては7割以上が期待していた程度に学べたとの回答
- プログラミングに関しては全体では56%程度が期待した程度に学べたと回答していますが、エントリークラスだけ見ると8割弱と高い割合となっています
- プロジェクトマネジメントについては半数程度が期待程度に学んでいます



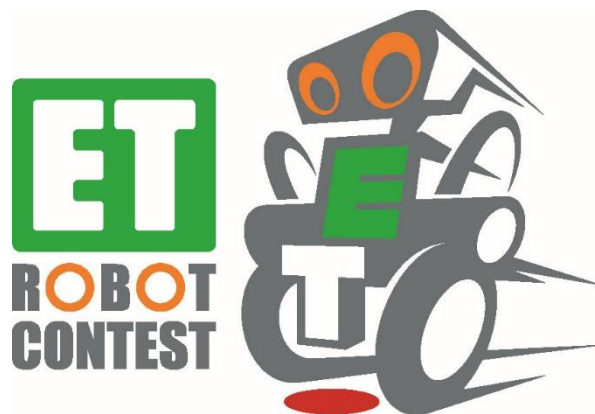
アンケートの声

- ・コロナで急なりモート開催になったにも関わらずこのような環境を提供していただいたのはありがたいと思います
- ・モデル図に対する指摘のコメントなど、今後の開発に参考になる点が多かったので非常に助かった
- ・ETロボコンはモデリングからコーディングまでの一連の流れを学べるよい機会でした
- ・ロボコン自体は初めてでしたが、非常に楽しませてもらいました。運営様の迅速な対応（特にslack）を見ていましたが、どれも分かりやすい回答かつ、有益な情報で非常に良かったです。

UMTP認定試験割引



- UMTP様のご協力により、UMTP認定試験 Level1-2を特別価格での提供を予定しています
 - 金額・人数などは決まり次第展開します
 - https://umtp-japan.org/about_exam/work_gaiyo
 - Level2を受講にするにはその下のレベルに合格している必要があります



ETロボコン2021 モデル審査

ETロボコン実行委員会