



エージェントシステム プラットフォーム

自動運転に必要な「危険予知エージェント」

優良なドライバーの特徴として危険予知能力が高いことがあります。危険予知エージェントを搭載することで危険予知能力を取り入れることができます。

どのようにエージェントを構築するのか？

エージェントの知能部分をモデルベースで開発します。具体的には決定表を用いて知能をルールとして表現します。

例えば、図1は危険予知エージェントがドライバーにヘッドアップディスプレイ(HUD)で雪道での危険予知を警告している様子を示しています。雪道運転での知識を決定表でモデリングしたものが表1です。



図1 危険予知エージェントシミュレーション画面

わだちルール	Y	Y	Y	N	Y	Y	Y
対向車	Y	Y	Y		N		
車速	> 40	> 40	> 40			<= 40	
TTC	>= 3	< 3	< 3				
運転手特性	== SNOW_INEXPERIENCE		== SNOW_INEXPERIENCE		== SNOW_EXPERIENCE		== SNOW_UNDEF
メッセージ1	わだち注意	対向車と同じわだち	対向車と同じわだち	わだち空	わだち空	わだち空	わだち空
メッセージ2	速度注意	衝突の危険	衝突の危険	衝突空	衝突空	衝突空	衝突空

表1 危険予知ルールモデル

▶危険予知エージェント動画URL

http://www.zipc.com/product/ZIPC_R&B/video/

<https://youtu.be/aaBo6BuVRts>



ルールベースをモデルベースにすることで従来の課題を解決

従来のエキスパートシステムで用いていたIF-THEN表記法では再利用、分割、階層化を行う事が難しく、ルールスパゲティ、ルール爆発を起こしていました。そこで、ルールにモデルベースを導入することでこれらの課題を解決し、マルチエージェントの構築もしやすくなりました(図2)。

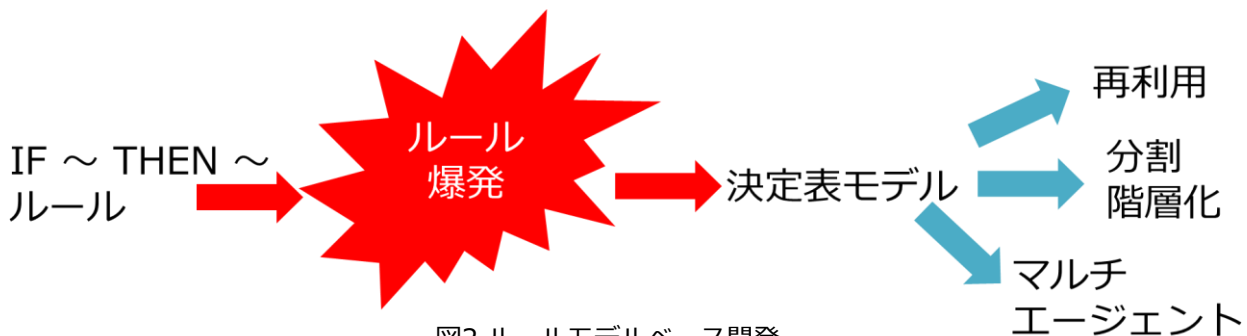


図2 ルールモデルベース開発

(*)決定表の書式はSpecification of single-hit decision table(決定表) ISO5086 (JIS X0125)に対応
(*) TTC(Time-To-Collision)

～ お気軽にお問い合わせを！ ▶▶▶ info@zipc.com ～



エージェントシステム プラットフォーム

様々な環境と連携したエージェント開発

様々な環境と連携してエージェント開発を行えます。

図3はゲームエンジンUnity上で自動運転ルールエージェントを動作させた例です。

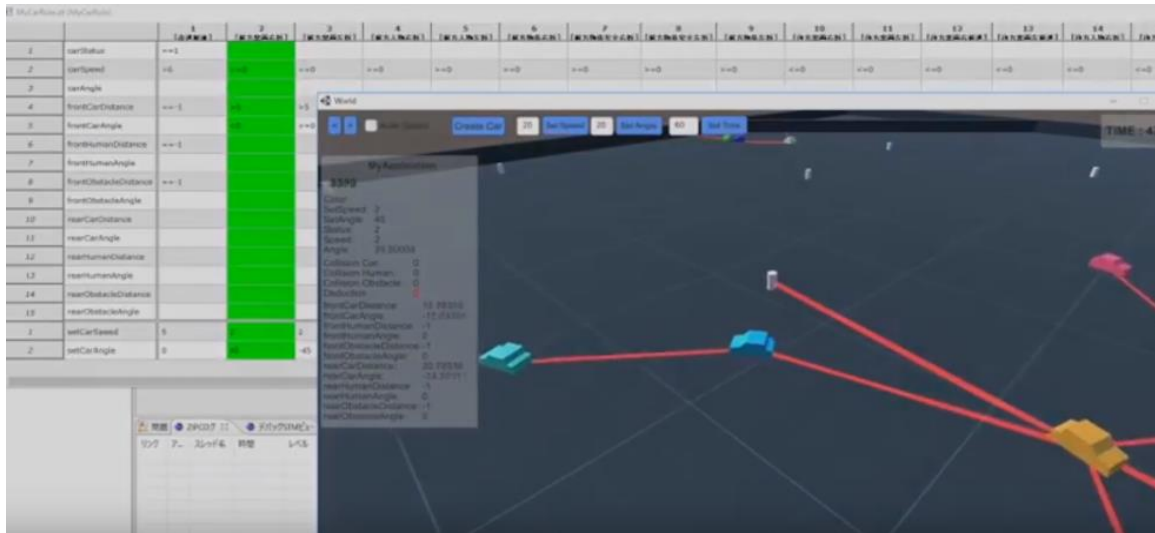


図3 ゲームエンジン Unity上での動作

▶判断エージェント動画URL

http://www.zipc.com/product/ZIPC_R&B/video/

<https://youtu.be/XgNlmpGhzVo>



様々なドメインのエージェント開発

ROS(Robot Operating System)などの環境と協調してエージェント開発を行えます。

図4はヒューマノイドロボット“NAO”の開発環境“Choregraphe”とZIPC Designer R&Bを用いて“NAO”に横断歩道を歩行する判断エージェントを構築した例です。

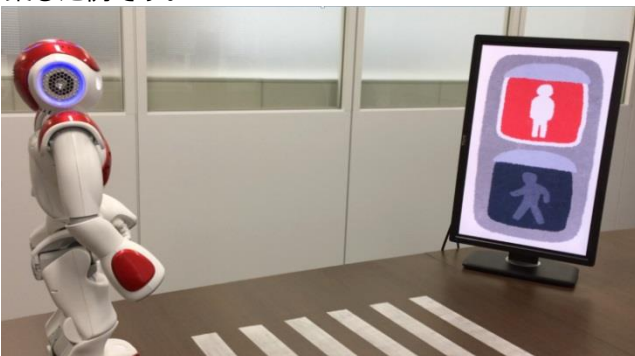
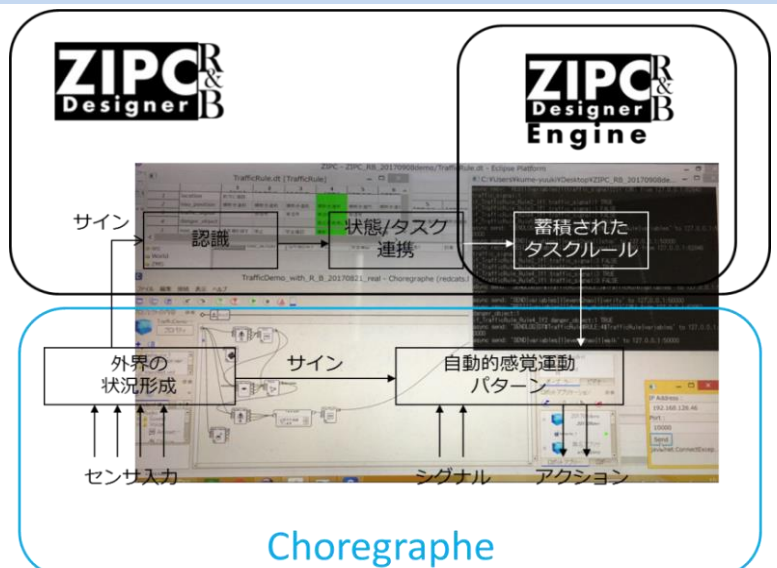


図4 ヒューマノイドロボットエージェント

▶判断エージェント動画URL

http://www.zipc.com/product/ZIPC_R&B/video/

<https://youtu.be/HuOkGi3AS0A>



- 自動運転システムとルールベースシステムに関する記事
http://www.nttdata.com/jp/ja/insights/trend_keyword/2016052701.html
- エッジインテリジェントを実現するルールベースシステム～自動運転システムの場合～
<http://www.zipc.com/instance/files/vol19/vol19-13.pdf>