

团体标准

T/ITS 0293.3-2025

自主式交通系统 交通语义表示语言 第3部分：语义信息交互

Autonomous transportation system —

Traffic semantic representation language—Part 3: Semantic information interaction

2025-11-26 发布

2025-11-26 实施

中国智能交通产业联盟 发布

中国智能交通产业联盟

目 次

前 言 II

1 范围 1

2 规范性引用文件 1

3 术语和定义 1

4 缩略语 2

5 交通语义信息交互框架 2

6 语义信息交互基本流程 3

7 语义信息交互消息 3

8 交通语义交互行为 7

附 录 A （资料性附录） 交通语义交互行为对应的消息体结构 8

附 录 B （资料性附录） 交通场景语义信息交互示例 12

参 考 文 献 20

前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中国智能交通产业联盟（C-ITS）提出并归口。

本文件主要起草单位：北京交通大学、北京邮电大学、交通运输部公路科学研究院、华路易云科技有限公司、联通智网科技股份有限公司、北京航空航天大学、株洲中车时代电气股份有限公司、交通运输部水运科学研究院。

本文件主要起草人：董宏辉、欧帆、王佳佳、袁泉、李静林、任毅龙、于海洋、李振华、余红艳、辛亮、于朝阳、王泉东、周昱诚、谌仪、潘小熙、兰征兴、顾惠楠、马攀科、林军、洪奕鹏、刘俊兰、李巍、周伟杰、吴昊、江培源。

自主式交通系统 交通语义表示语言 第3部分：语义信息交互

1 范围

本文件规定了交通语义表示语言的语义信息交互，包括语义信息交互框架、语义信息交互流程、语义信息交互消息以及交通语义交互行为。

本文件适用于采用交通语义表示语言的主体在交互过程中实现语义交互，包括但不限于人、载运装备、基础设施和交通管控中心等交通主体间的语义交互场景。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

T/CSAE 53—2020 合作式智能运输系统 车用通信系统应用层及应用数据交互标准（第一阶段）

T/ITS 0098-2017 合作式智能运输系统 通信架构

T/ITS 0292-2025 自主式交通系统 互操作机制模型

T/ITS 0293.1-2025 自主式交通系统 交通语义表示语言 第1部分：通用定义

T/ITS 0293.2-2025 自主式交通系统 交通语义表示语言 第2部分：语法规则

3 术语和定义

T/ITS 0292-2025、T/ITS 0293.1-2025界定的以及下列术语和定义适用于本文件。

3.1

自主式交通系统 autonomous transportation system

以自主感知、自主决策、自主执行为特征的高度智能、高度自治的交通系统。

[来源：T/ITS 0292-2025]

3.2

自主式交通主体 autonomous traffic agent

自主式交通主体是能够在复杂交通环境下，独立完成感知、认知、决策与控制闭环，实现预定交通任务的交通智能体。

[来源：T/ITS 0293.1-2025]

3.3

交通语义表示语言 traffic semantic representation language

交通语义表示语言是一种以形式化方式准确描述交通内容的语言，具备语义表示、语义理解、语义交互、逻辑推理和互操作等能力。

[来源：T/ITS 0293.1-2025]

3.4

语义信息交互 semantic information interaction

人、载运装备、基础设施、交通管控中心这四类交通主体通过交互框架传递语义信息，实现可理解的信息交互。

4 缩略语

RSU：路侧单元（Road Side Unit）

TSRL：交通语义表示语言（Traffic Semantic Representation Language）

5 交通语义信息交互框架

本文件关注的是交通语义信息交互框架、交互流程、交互消息的层次化结构以及交通语义交互行为。交通语义信息交互框架如图1所示。交通语义信息交互流程和交互消息，对应于消息层进行语义信息交互的基本流程和消息的结构形式，可承载符合第1、2部分规范的语义表示内容本文件不指定底层的通信方式，可以用于各种不同的网络层、接入层技术，帮助异构交通主体在语义层面实现统一的信息交互。

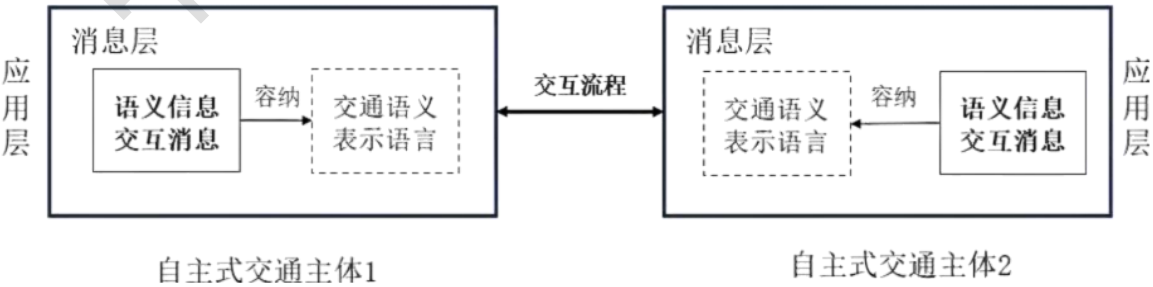
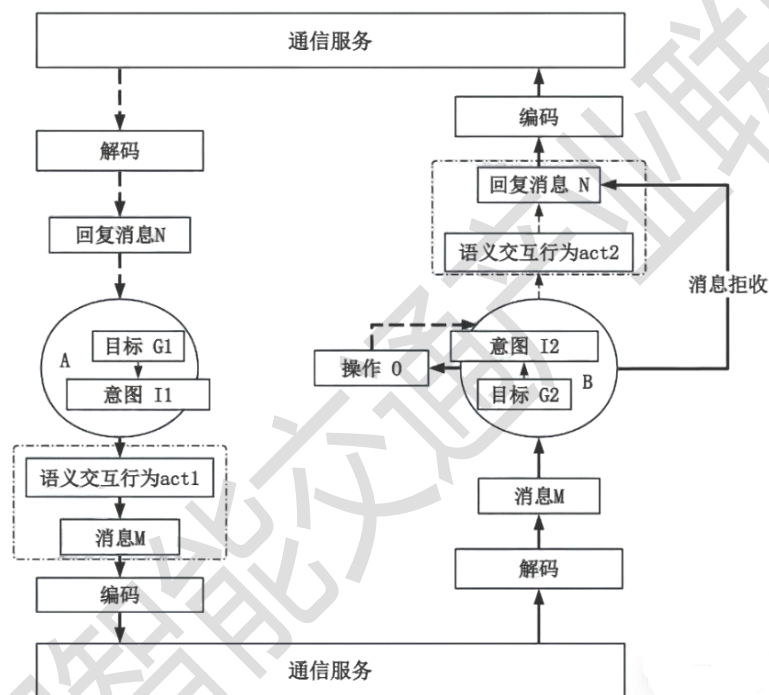


图1 交通语义信息交互框架

交互消息遵循“消息帧-消息体-消息参数”层层嵌套的逻辑进行制定。

6 语义信息交互基本流程

图2说明了多交通主体进行语义信息交互应满足的基本流程。图2包含两个交通主体：交通主体A和交通主体B，其中A是会话的发起者，它先发送消息。图2显示交通主体语义交互的总体过程：交通主体A期望达成目标G1，因此形成特定意图I1，决定采取语义交互行为act1，将该行为形成消息M，并对消息M编码，使其满足底层通讯协议与服务的要求，传输给交通主体B。若交通主体B选择拒收消息，则会直接回复拒收消息；若B选择接收消息，则根据事先定义的规范以及自身目标G2与意图I2，采取语义交互行为act2并选择是否回复消息N，同时也可能执行某些操作O，此后重复这一流程。



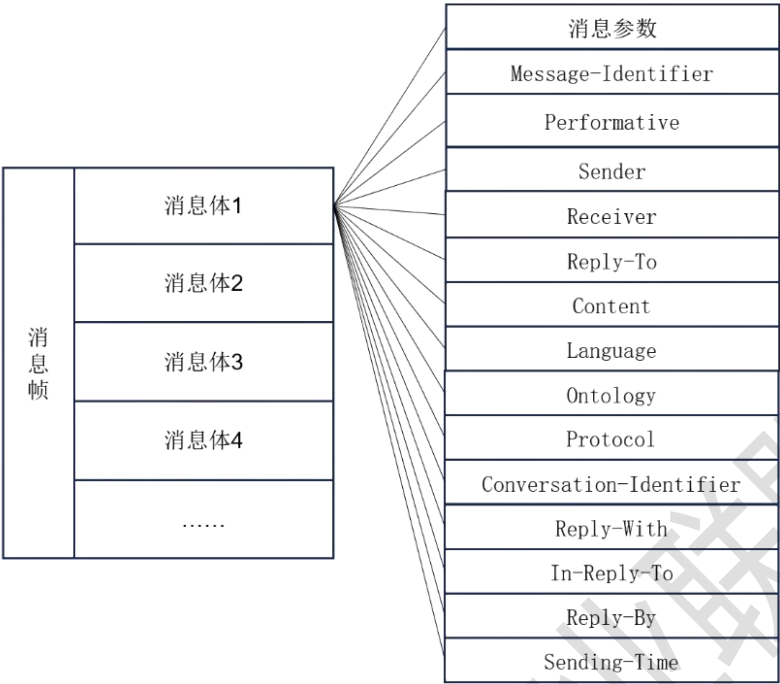


图3 语义信息交互消息层次结构示意图

7.2 消息帧

一个消息帧可以容纳一个或多个消息体。消息帧结构形式如表1所示。消息体包含一系列参数，如表2。消息参数中消息内容Content的具体实现不在本文件的范围之内。

表1 消息帧结构形式

类型	表示形式	说明
消息帧	<pre>MessageFrame ::= SEQUENCE { Message1 Message2 Message3 ... }</pre>	MessageFrame 为消息帧名；SEQUENCE 表示后续内容依次序，“{}”中为包含的消息体；Message1 等为消息体名；消息体的内容在此处省略。

7.3 消息体

消息体用于传输语义消息，其结构规范如表2所示：

表2 消息体结构规范表

类型	表示规范	说明
消息体	<pre> MessageName ::= SEQUENCE{ Message- Identifier (<number>) Performative (<performative>) : Sender (<subject>) : Receiver (<subject>) : Reply-To (<subject>) : Content “<expression>” : Language (<language>) : Ontology (<ontology>) : Protocol (<protocol>) : Conversation- Identifier (<number>) : Reply-With (<expression>) : In-Reply-To (<expression>) : Reply-By (<time>) : Sending-Time(<timestamp>) } </pre>	MessageName 为消息体名，“{}”中为包含的消息参数，消息参数的内容由< >中的对象表示。

消息体应包含一个或多个消息参数，如表2。消息体的类别与所需参数的种类及个数由参数“Performative”决定，该参数实质上也代表了语义交互行为类型。

消息体参数支持扩展。除了可以使用表3中指定的消息参数之外，在面向具体实现时，用户可以自由地定义消息体参数。这些非标准附加参数的名称必须使用前置字符串“X-”。

表3 消息体参数表

参数类别	参数	说明
消息体的名称	Message-Identifier	表示消息体标识符。
交互的行为类型	Performative	表示消息的类型，对应于语义消息交互的行为类型。
交互的参与者	Sender	表示消息的发送者。
	Receiver	表示消息的预期接收者的身份。
	Reply-To	表示后续消息的回复对象。
消息内容	Content	表示消息的内容，为交互语句。
内容描述	Language	表示消息语言，表达参数“Content”消息内容所使用的语言。
	Ontology	表示用于赋予参数“Content”消息内容中的符号含义的个体。
交互控制	Protocol	表示通信协议
	Conversation-Identifier	表示会话标识符。

表3 消息体参数表（续）

参数类别	参数	说明
交互控制	Reply-With	表示响应标识符。
	In-Reply-To	表示回复标识符。
	Reply-By	表示发送消息发出时与回复消息发出时能接受的最大时间间隔。
	Sending-Time	表示发送者发出消息的时间。

各个参数的具体说明如下：

- a) Message-Identifier。消息体标识符。用于唯一地表示消息帧中的消息体。可以是数字或某种编码。“Message-Identifier”是所有交互消息的必需参数；
- b) Performative。述行词。表示消息交互的行为类型，同时也体现出消息的类型。“Performative”是所有交互消息的必需参数。此外，不同的述行词决定了消息体的参数数量与类型；
- c) Sender。发送者。表示消息发送者的身份。“sender”参数是大多数消息的参数。但是，如果发送消息的主体希望保持匿名，则可以省略发送者参数；
- d) Receiver。接收者。表示消息的预期接收者的身份。“Receiver”参数可以是单个交通主体或多个主体的非空集合。后者对应于消息多播的情况；
- e) Reply-To。消息回复对象。该参数指示此会话中的后续消息将定向到“Reply-To”参数中指定的主体，而不是发送到“Sender”参数中指定的主体；
- f) Content。消息内容。表示消息的内容，包含交通语义表示语言的语句或表示交通语义的其它数据形式。任何消息的内容含义均由消息的接收方进行解析；
- g) Language。消息语言。表示表达参数“Content”消息内容所使用的语言。如果已知接收消息的主体知道表达消息内容所使用的语言，则可以省略该字段。
- h) Ontology。本体。表示用于赋予参数“Content”消息内容中的符号含义的本体。当发送者与接收者都已知该参数值时，可省略。在交通领域，“Ontology”可指不同交通方式下各类符号及其含义的本体。例如，道路交通本体“RoadTraffic”，轨道交通主体“RailTransit”，水运交通本体“WaterTransportation”，它们应定义各自交通方式中各类符号及其含义；
- i) Protocol。通信协议。表示发送代理在消息中采用的底层通信协议；
- j) Conversation-Identifier。会话标识符。表示正在进行的消息序列所属哪个会话。交通主体可以使用会话标识符来标记消息，以管理其通信策略和活动。允许交通主体与其它多个交通主体进行单独会话。需要对 Conversation-id 参数使用全局唯一值，以便参与者能够区分多个并发会话；
- k) Reply-With。响应标识符。表示响应主体将使用该表达式来识别此消息。“Reply-With”参数设计用于在多个对话同时发生的情况下跟踪对话线程。例如，如果主体 i 向主体 j 发送一条消

息，其中包含：

Reply-With <expression>

主体j将回复一条消息，其中应包含：

In-Reply-To <expression>:

- l) In-Reply-To。回复标识符。表示该消息为此前较早消息的回复消息。用法如上所示；
- m) Reply-By。最长答复间隔。表示发送消息发出时与回复消息发出时能接受的最大时间间隔, 单位为毫秒。
- n) Sending-Time。消息发送时间。表示发送者发出消息的时间，采用时间戳表示自 1970 年以来的毫秒数。

8 交通语义交互行为

8.1 语义交互行为类型

在不同的交通方式下，交通主体参与的交通场景有很大的差异，但这些场景大多需要人、载运装备、基础设施与交通管控中心进行协同配合才能实现。从人-载运装备-基础设施-交通管控中心之间进行信息交互的场景中，抽象出交通主体进行信息交互的动作行为，分为信息交互、协作控制、错误处理三类交互类型，包括告知、查询、请求、重复执行、接受、拒绝、失败等行为，具体对应于告知消息、查询消息、请求消息、重复执行消息、接受消息、拒绝消息、失败消息、困惑消息等语义消息，如表 4 所示。

表4 语义消息分类表

语义消息类型	行为类型	交互类型
告知消息	告知	信息交互
查询消息	查询	
请求消息	请求	协作控制
重复执行消息	重复执行	
接受消息	接受	
拒绝消息	拒绝	
失败消息	失败	错误处理
困惑消息	困惑	

8.2 对应的语义消息结构

对信息交互、协作控制以及错误处理这三类语义信息交互类型中包含的交互行为进行结构语法定义。这些交互行为对应于消息参数“Performative”的值，并决定了消息体的结构，即其需要的参数种类及个数。在附录A中，对每类交互行为对应消息体的结构做了详细说明。在附录B中，对道路、轨道、水运三种交通方式下应用该方法进行交通主体语义信息交互做了详细说明。

附 录 A
(资料性附录)
交通语义交互行为对应的消息体结构

A.1 信息交互

- a) Inform。告知行为。该行为表示发送者告知接收者某些信息。消息体结构如下：

```
MessageName {  
    Message-Identifier (<number>)  
    Performative (Inform)  
    : Sender (<subject>)  
    : Receiver (<subject>)  
    : Content  
    “<expression>”  
    : Conversation-Identifier (<number>)  
    : In-Reply-To (<expression>)  
    : Sending-Time (<timestamp>)  
}
```

- b) Query。询问行为。该行为表示发送者希望接收者答复某些信息。消息体结构如下：

```
MessageName {  
    Message-Identifier (<number>)  
    Performative (Query)  
    : Sender (<subject>)  
    : Receiver (<subject>)  
    : Reply-To (<subject>)  
    : Content  
    “<expression>”  
    : Conversation-Identifier (<number >)  
    : Reply-With (<expression>)  
    : Reply-By (<time>)  
    : Sending-Time (<timestamp>)  
}
```

A.2 协作控制

- a) Request。请求行为。表示发送者请求接收者执行某些行为。消息体结构如下：

```

MessageName {
    Message-Identifier (<number>)
    Performative (Request)
    : Sender (<subject>)
    : Receiver (<subject>)
    : Content
    “<expression>”
    : Conversation-Identifier (<number >)
    : Reply-With (<expression>)
    : Reply-By (<time>)
    : Sending-Time (<timestamp>)
}

```

- b) Request-whensoever。重复执行。表示发送者希望接收者在某个命题变为真时立即执行某些操作，此后每次该命题再次变为真时，再次执行同样操作。消息体结构如下：

```

MessageName {
    Message-Identifier (<number>)
    Performative (Request-whensoever)
    : Sender (<subject>)
    : Receiver (<subject>)
    : Content
    “<expression>”
    : Conversation-Identifier (<number >)
    : Reply-With (<expression>)
    : Reply-By (<time>)
    : Sending-Time (<timestamp>)
}

```

- c) Accept。接受行为。表示发送者接受先前收到的要求并以成功执行相关操作。消息体结构如下：

```

MessageName {
    Message-Identifier (<number>)
    Performative (Accept)
    : Sender (<subject>)
    : Receiver (<subject>)
    : Content
}

```

```

    “<expression>”
    : Conversation-Identifier (<number >)
    : In-Reply-To (<expression>)
    : Sending-Time (<timestamp>)
  }

```

- d) Refuse。拒绝行为。表示发送者拒绝执行某些行为并给出理由。消息体结构如下：

```

MessageName {
  Message-Identifier (<number>)
  Performative (Refuse)
  : Sender (<subject>)
  : Receiver (<subject>)
  : Content
  “<expression>”
  : Conversation-Identifier (<number >)
  : In-Reply-To (<expression>)
  : Sending-Time (<timestamp>)
}

```

A.3 异常处理

- a) Failure。失败行为。表示发送者告知接收者其已尝试执行某项操作但尝试失败。消息体结构如下：

```

MessageName {
  Message-Identifier (<number>)
  Performative (Failure)
  : Sender (<subject>)
  : Receiver (<subject>)
  : Content
  “<expression>”
  : Conversation-Identifier (<number >)
  : In-Reply-To (<expression>)
  : Sending-Time (<timestamp>)
}

```

- b) Confuse。困惑行为。表示发送主体无法理解接收信息的内容。消息体结构如下：

```
MessageName {  
    Message- Identifier (<number>)  
    Performative (Confuse)  
    : Sender (<subject>)  
    : Receiver (<subject>)  
    : Content  
    “<expression>”  
    : Conversation-Identifier (<number >)  
    : In-Reply-To (<expression>)  
    : Sending-Time (<timestamp>)  
}
```

附 录 B
(资料性附录)
交通场景语义信息交互示例

B.1 道路交通场景语义信息交互示例

B.1.1 道路交通场景描述

在高速公路上行驶的车辆，可能需要进行车道变换以适应不同的行驶要求，比如超车、驶出高速公路或避让慢速行驶的车辆等。车辆变道时需要考虑周围车辆的位置和速度，以确保变道过程中的安全通行。此外，车辆变道可能受到最低限速、最大限速和道路出口位置等因素的影响，因此需要根据具体情况采取适当的行驶策略。这些信息可通过车-车以及车-路侧设施之间的语义信息交互获取。

由于本文件并不指定具体的底层通信协议同时也不确定消息发出时间，此处省略参数“Protocol”与“Sending-Time”。

对于道路场景的信息定义以及场景知识表达参照标准《自主式交通系统 交通语义表示语言 第1部分：通用术语》与《自主式交通系统 交通语义表示语言 第2部分：语法规则》。参数“Language”的值即为交通语义表示语言，记为TSRL，即Traffic Semantic Representation Language，参数“Ontology”为系列标准第1部分定义的道路交通本体库，记为RoadTraffic。

B.1.2 道路交通主体语义信息交互具体实现

a) 车车交互。

车辆Vehicle1向车辆Vehicle2询问Vehicle2的位置、速度、距离高速公路出口的距离等信息。

```
QueryMessageV2V {
  Message- Identifier (1)
  Performative (Query)
  : Sender (Vehicle1)
  : Receiver (Vehicle2)
  : Content
    “InLane(Vehicle2, Data),
    HasSpeed(Vehicle2, Data),
    DistanceToExit(Vehicle2, Data),
    ...”
  : Language (TSRL)
  : Ontology (RoadTraffic)
  : Conversation-Identifier (1)
  : Reply-With (VehicleInfo01)
```


: Reply-By (1000ms)}

车辆Vehicle2向车辆Vehicle1答复Vehicle2的位置、速度、距离高速公路出口的距离等信息。

```
InformMessageV2V {
    Message- Identifier (2)
    Performative (Inform)
    : Sender (Vehicle2)
    : Receiver (Vehicle1)
    : Content
    "InLane(Vehicle2, Data),
    HasSpeed(Vehicle2, Data),
    DistanceToExit(Vehicle2, Data),
    ..."
    : Language (TSRL)
    : Ontology (RoadTraffic)
    : Conversation-Identifier (1)
    : In-Reply-To (VehicleInfo01)
    : Reply-By (1000ms)
}
```

车辆Vehicle1请求车辆Vehicle2减速。

```
RequestMessage{
    Message-id (3)
    Performative (Request)
    : Sender (Vehicle1)
    : Receiver (Vehicle2)
    : Content
    "Let(Vehicle2, SpeedDown)"
    : Conversation-id (1)
    : Reply-With (VehicleQuest01)
    : Reply-By (1000ms)
}
```

车辆Vehicle2接受车辆Vehicle1的减速请求。

```
AcceptMessage {
    Message-id (4)
    Performative (Accept)
    : Sender (Vehicle2)
```

```

: Receiver (Vehicle1)
: Content
  "Let(Vehicle2, SpeedDown)"
: Conversation-id(1)
: In-Reply-To (VehicleQuest01)
}

```

b) 车路交互

路侧单元RSU向周围车辆Vehicle1, Vehicle2, Vehicle3等交通环境数据, 包括道路条件、天气状况、能见度等。

```

InformMessageV2R {
  Message-Identifier (3)
  Performative (Inform)
  : Sender (RSU)
  : Receiver (Vehicle1, Vehicle2, Vehicle3, ...)
  : Content
    "HasRoadCondition(Road, Condition),
     WeatherCondition(Weather),
     Visibility(Visibility),
     ..."
  : Language (TSRL)
  : Ontology (RoadTraffic)
  : Conversation-Identifier (2)
  : Reply-With (RoadCondition01)
  : Reply-By (1000ms)
}

```

车辆Vehicle向路侧单元RSU查询当前道路Road的道路数据, 包括道路限速等信息。

```

QueryMessageV2R {
  Message-Identifier (4)
  Performative (Query)
  : Sender (Vehicle)
  : Receiver (RSU)
  : Content
    "MaxSpeed (Road, Data),
     MinSpeed (Road, Data),
     ..."
  : Language (TSRL)
  : Ontology (RoadTraffic)
  : Conversation-Identifier (2)
  : Reply-With (RoadData01)
}

```

```

: Reply-By (1000ms)
}

```

路侧单元RSU向车辆Vehicle答复当前道路Road的道路数据，包括道路限速等信息。

```

InformMessageR2V {
  Message-Identifier (5)
  Performative (Inform)
  : Sender (RSU)
  : Receiver (Vehicle)
  : Content
  "MaxSpeed (Road, Data),
   MinSpeed (Road, Data),
   ..."
  : Language (TSRL)
  : Ontology (RoadTraffic)
  : Conversation-Identifier (2)
  : In-Reply-To (RoadData01)
  : Reply-By (1000ms)
}

```

B.2 轨道交通场景语义信息交互示例

B.2.1 轨道交通场景描述

选取城市轨道交通中基于道岔的列车冲突消解场景。在一个特定的道岔，考虑两辆列车A与B接近并试图通过这个道岔。轨道控制中心的调度系统检测到两辆列车即将在一个道岔区段产生路径冲突。在此场景中，控制中心为列车A优先分配通行权，指令列车B在进路前停车等待。系统将决策转化为具体的控制命令，包括：锁定道岔为列车A所需位置、将列车B进路信号设置为禁止、并向列车B发送停车建议或指令。系统持续监控列车B的减速与停车状态，以及列车A的顺利通过状态，直至冲突风险完全解除。

由于本文件并不指定具体的底层通信协议同时也不确定消息发出时间，此处省略“Protocol”与“Sending-Time”。

对于轨道场景的信息定义以及场景知识表达参照标准《自主式交通系统 交通语义表示语言 第1部分：通用术语》与《自主式交通系统 交通语义表示语言 第2部分：语法规则》。参数“Language”的值即为该标准定义的语言，记为TSRL，即Traffic Semantic Representation Language。参数“Ontology”为该标准定义的轨道交通本体库，记为RoadTransit。

B.2.2 轨道交通主体语义信息交互具体实现

a) 控制中心与转辙机交互。

控制中心要求转辙机变更道岔位置，使列车 A 优先通行。

```

QuestMessageControlCenter2Switch {
    Message-Identifier (1)
    Performative (Quest)
    : Sender (ControlCenter)
    : Receiver (Switch)
    : Content
    "Let(Switch, State1),
    ..."
    : Language (TSRL)
    : Ontology (RailTransit)
    : Conversation-Identifier (1)
    : Reply-With (ControlCenterQuest01)
    : Reply-By (1000ms)
}

```

b) 控制中心与信号设备交互。

控制中心要求将列车 B 进路信号设置为禁止。

```

QuestMessageControlCenter2Signal {
    Message-Identifier (2)
    Performative (Inform)
    : Sender (ControlCenter)
    : Receiver (Signal)
    : Content
    "LetSignalState(Signal, State2),
    ..."
    : Language (TSRL)
    : Ontology (ControlCenterTransit)
    : Conversation-Identifier (2)
    : Reply-With (TrainQuest02)
    : Reply-By (1000ms)
}

```

c) 控制中心与列车交互。

控制中心并向列车 B 发送停车建议或指令。

```

QuestMessageControlCenter2Train {
    Message-Identifier (4)
    Performative (Quest)
    : Sender (ControlCenter)
    : Receiver (TrainB)
    : Content
    "Let (TrainB, Stop),
    ..."
    : Language (TSRL)
    : Ontology (RailTransit)
    : Conversation-Identifier (3)
    : Reply-With (ControlCenterQuest03)
    : Reply-By (1000ms)
}

```

列车 B 向控制中心发送自身的状态信息。

```

InformMessageST2DC {
    Message-id (5)
    Performative (Inform)
    : Sender (TrainB)
    : Receiver (ControlCenter)
    : Content
    "HasSpeed (TrainB, Speed),
    ..."
    : Conversation-id(3)
    : In-Reply-To (DispatchCenterQuest01)
}

```

B.3 水运交通场景语义信息交互示例

B.3.1 水运交通场景描述

选取水运交通场景，考虑两艘船在航道上相遇的情况。航道可以是海洋、河流或其他水域中的指定航行通道。船舶A和船舶B在航道上沿着各自的航线航行，这导致它们在某一时刻会在航道上相遇或交叉。由于航道空间有限，这种相遇可能会带来潜在的碰撞风险。因此，在船舶相遇时，必须采取适当的航行规则和安全措施，以确保船舶之间的安全通行。在这一过程中，船舶的速度、航向和位置都是动态变化

的，船舶之间的距离也会随着时间变化而变化。船舶相遇时的航行决策需要考虑诸多因素，包括航行规则、船舶动态特性以及海上交通情况，以最大程度地降低碰撞风险，并保障水运交通的安全和效率。

由于本文件并不指定具体的底层通信协议同时也不确定消息发出时间，此处省略参数“Protocol”与“Sending-Time”。

对于水运场景的信息定义以及场景知识表达参照标准《自主式交通系统 交通语义表示语言 第1部分：通用术语》与《自主式交通系统 交通语义表示语言 第2部分：语法规则》。参数“Language”的值即为该标准定义的语言，记为TSRL，即Traffic Semantic Representation Language。参数“Ontology”为该标准定义的水运交通本体库，记为WaterTransportation。

B.3.2 水运交通主体语义信息交互具体实现

a) 船舶交互。

船只 Ship1 向船只 Ship2 询问 Ship2 的位置、速度等信息。

```
QueryMessageS2S {
    Message-Identifier (1)
    Performative (Query)
    : Sender (Ship1)
    : Receiver (Ship2)
    : Content
    "HasPosition (Ship2, Data),
    HasSpeed (Ship2, Data),
    ..."
    : Language (TSRL)
    : Ontology (WaterTransportation)
    : Conversation-Identifier (1)
    : Reply-With (ShipQuery01)
    : Reply-By (1000ms)
}
```

船只 Ship2 向船只 Ship1 答复 Ship2 的位置、速度、等信息。

```
InformMessageS2S {
    Message-Identifier (2)
    Performative (Inform)
    : Sender (Ship2)
    : Receiver (Ship1)
    : Content
```

```

    “HasPosition(Ship2, Data),
      HasSpeed(Ship2, Data),
      ...”
  : Language (TSRL)
  : Ontology (WaterTransportation)
  : Conversation-Identifier (1)
  : In-Reply-To (ShipQuery01)
  : Reply-By (1000ms)
}

```

b) 船只与航道中的传感设备交互。

航道中的传感设备 SensingDevice 向船只 Ship1、Ship2 等广播航道长度、宽度、天气状况等信息。

```

InformMessageSD2S {
  Message-Identifier (3)
  Performative (Inform)
  : Sender (SensingDevice)
  : Receiver (Ship1, Ship2, ...)
  : Content
    “HasChannelLength(Channel, Data),
      HasChannelWidth(Channel, Data),
      WeatherCondition(Data),
      ...”
  : Language (TSRL)
  : Ontology (WaterTransportation)
  : Conversation-Identifier (2)
  : In-Reply-To (ChannelInfo01)
  : Reply-By (1000ms)
}

```

参 考 文 献

- [1] FIPA SC00061G 智能体通信语言消息结构规范 (ACL Message Structure Specification)

中国智能交通产业联盟

中国智能交通产业联盟

标准

自主式交通系统 交通语义表示语言 第3部分：语义信息交互

T/ITS 0293.3-2025

北京市海淀区西土城路8号（100088）

中国智能交通产业联盟印刷

网址：<http://www.c-its.org.cn>

2025年11月第一版 2025年11月第一次印刷