arXiv:2302.06970v1 [cs.AI] 14 Feb 2023

超媒体多智能体系统中的一等抽象：指示符

Danai Vachtsevanou 圣加仑
大学 圣加仑，瑞士
danai.vachtsevanou
@unisg.ch

Andrei Ciortea 圣加仑
大学 圣加仑，瑞士
andrei.ciortea
@unisg.ch

Simon Mayer 圣加仑
大学 圣加仑，瑞士
si
mon.mayer@unisg.ch

Jérémy Lemée
圣加仑大学 圣加仑，瑞士 j
eremy.lemee@unisg.ch

摘要

超媒体 API 支持设计可复用的超媒体客户端，使其能够在万维网上发现并利用可供性。然而，这类客户端的可复用性仍然有限，因为它们无法规划交互，也无法对交互进行推理。本文在万维网上由超媒体驱动的可供性利用，与多智能体系统（MAS）乃至更广泛的人工智能领域中已被广泛探索的动作表示与推理方法之间，搭建了一座概念桥梁。我们借鉴可供性理论和人机交互领域中的概念与方法，这些概念与方法有助于提升开放且可演化环境中的交互效率；据此，我们将示符引入为基于万维网的多智能体系统中的一等抽象：示符根据其使用时所处的智能体—环境上下文进行设计，使具备异质能力的智能体能够采取动作并对动作进行推理。我们为超媒体环境中示符的上下文暴露定义了一种形式模型，旨在推动可供性利用。我们通过一个基于万维网的多智能体系统原型展示了这一方法：其中，两个推理能力不同的智能体只需感知与自身能力相匹配的示符，便能主动发现如何与环境交互。我们表明，示符暴露可以基于动态的智能体—环境关系进行内在管理

t

上下文，以促进万维网上有效且高效的交互。

关键词

自主超媒体客户端、超媒体多智能体系统、指示符、可供性理论

1 引言

关于自主智能体和多智能体系统（MAS）的研究，一直聚焦于灵活利用环境资源的模型和方法。为了构建大规模且富含可供性的多智能体系统，研究者将万维网视为一种赋能手段，使自主智能体能够充分发挥其推理和决策能力，为实现复杂目标而发现并利用万维网资源的可供性 [9]。在万维网上发现和利用可供性的主要机制是超媒体驱动的交互，这种交互基于对高层语义信息 与低层控件 [11]：智能体可以根据语义信息进行推理，据此决定采取何种动作，例如，某件物品靠近机械臂，因此可以抓取该物品；随后，智能体通过使用配套控件执行动作，例如向机械臂的 Web API 发送 HTTP 请求。执行动作后，根据万维网架构的“超媒体作为应用状态引擎”（HATEOAS）原则 [10]，，会出现新的超媒体，使智能体能够推理并利用新的可供性，例如拾取已抓取的物品。

W3C 物联网 [6] 已通过超媒体驱动 API 的机器可读规范扩展了万维网环境，即 事物描述。这些规范使应用程序能够构建于领域特定抽象之上，而非底层控制之上（例如，针对某种特定应用层协议，如 HTTP）。因此，这类应用程序成为可复用的超媒体客户端，能够利用万维网上不同物理和虚拟实体提供的可供性。然而，即便是这类客户端，其行为也更像实用工具应用程序，而非超媒体的自主用户，因为其缺乏适应性，限制了可复用性：客户端会搜索特定语义信息，并在发现这些信息后，根据某些预先编译的逻辑触发对控制项的利用；这类似于一个智能体，像条件反射一样，总是选择附近的物品。

这与人类用户形成对比：他们能够自主地且灵活地主动采取行动，通过探索环境以及基于示符进行解释、推理和规划，利用新的可供性并动态实现自己的目标。示符是代表可供性高级语义信息的线索，专门依据人机交互（HCI）的原则设计，旨在提升可供性的可解释性和可发现性 [22]，，并以补充运行时人类—环境情境的方式呈现。因此，即使身处全新且充满可供性的环境，人类用户也能有效且高效地灵活发现和利用可供性。

因此，与人类相比，当今的超媒体客户端缺少：a）对示符等实体的认知，而这些实体本可策略性地引导它们在万维网环境中的交互（例如通过超媒体作为应用状态引擎）；以及 b）基于感知到的此类环境实体，对自身动作进行推理、规划并主动审慎思考的能力。

本文提出将示符作为基于超媒体的多智能体系统中的一等抽象，为能够进行推理和行动的客户端提供机器可读的交互规范，

使其能够作为万维网上的自主智能体运行。受人机交互的交互原则启发，我们致力于以用户为中心地设计示符，使其能够直接适应超媒体环境以及自主智能体的多样性，并满足其能力和目标。具体而言，我们定义了一种机制的形式模型，用于在超媒体环境中根据上下文呈现示符，以基于超媒体作为应用状态引擎提升交互的有效性和效率。

为了展示我们的方法，我们实现了一个原型基于超媒体的多智能体系统，其中两个具有不同推理能力的智能体通过动态感知适合自身能力和上下文的示征，主动发现如何与环境中的共享制品交互。我们展示了，面向自主智能体开放此类交互规范，可以根据动态智能体—环境上下文进行内在管理，从而促进在万维网上开展有效且高效的交互。

2 背景与相关工作

我们概述了相关研究，探讨人类智能体 （第 2.1 节）如何利用可供性，与超媒体客户端 （第 2.2 节）利用可供性的方式有何不同，以及关于多智能体系统 的研究如何用于缩小二者之间的差距（第 2.3 节）。

2.1 以人为中心的交互设计 ign

可供性理论 [15] 考察动物如何通过感知并利用环境中的可供性来控制自身行为。Chemero 将可供性 定义为一种行为可能性，它是以下两者之间的关系：a）智能体的能力 ； b）包含智能体和环境特征的情境 [7]。可供性的能力和情境被视为彼此互补 ，也就是说，在某个情境中存在具备某种能力的智能体，就使该智能体能够利用 某项可供性。例如，若人类智能体具备在手中无物情境下伸手够到某物品的能力，则该物品的可抓取可供性便可供其利用。文献 [7] 对可供性 aff的形式化定义如下：

$$aff =< a, s >, \tag{1}$$

其中， a是一种智能体能力， s是一种智能体—环境情境；两者同时存在，使智能体能够利用 aff。

当某种可供性变得可供智能体利用时，智能体便具有执行某种能力 ，以实施某种行为。例如，如果某件物品确实可抓取，那么人类代理就具有抓取该物品的能力，即实施抓取这一行为。因此，能力 $a_{\prime}$定义为智能体在可供性 aff可供利用时实施行为 b的特质；形式化定义如下：

$$a' =< aff, b >, \tag{2}$$

其中， aff是一种可供性（参见定义 1）， b是智能体在 aff可供利用时具有能力实施的行为。

能力 $a_{\prime}$反过来也可能与另一种情境$s_{\prime}$互补，使得 $a_{\prime}$和 $s_{\prime}$同时存在时，智能体便可利用另一种可供性 $aff_{\prime}$，即 $aff_{\prime} =< a_{\prime}s_{\prime} >$。例如，抓取能力可以被视为与「物品顶部没有任何东西」这一情境互补。这种互补关系可以通过可供性 $aff_{\prime}$来定义，即物品的可拾取可供性。

根据定义1和定义2，可供性与能力以彼此为基础进行非良基定义，从而形成一个可通过超集 [7]研究的复杂系统。为此，系统以可供性集和能力集建模，用于考察（人类）智能体如何控制系统中的行为：考虑系统中可执行行为的集合 B，$B = \{b_1, b_2, ..., b_n\}$（例如， $B = \{grasp, reach, open\text{-}door, ...\}$）。那么，针对 B所考虑的可供性和能力，分别包含在集合 $Affords\text{-}B$和 $Ability\text{-}to\text{-}B$中。这些集合以非良基方式相互定义如下：

$$Affords\text{-}B = \{< a_1, s_1 >< a_2, s_2 >, ..., < a_n, s_n >\}, \tag{3}$$

$$Ability\text{-}to\text{-}B = \{< aff_1, b_1 >, < aff_2, b_2 >, ..., < aff_n, b_n >\}^1. \tag{4}$$

可供性为人机交互（HCI）领域提供了宝贵的启发，推动人们设计人工制品，以提高其所提供可供性的可发现性 和可解释性 。具体而言，Norman 引入了指示符 这一概念，用以探讨如何设计可感知的线索，使其能够被有意义地解读，从而揭示有关可供性的信息 [25, 26]。具体而言，Norman 倡导以以人为本 的方式设计指示符，并以目标用户的需求、目标和能力为指导。由此，指示符成为环境线索，人们能够直观且可靠地 发现和解读这些线索，从而了解可能有哪些行为 ，以及可以如何 实施这些行为。

对环境中示符的系统化设计与呈现，有助于人类智能体探索、推理并利用可供性，即使身处规模庞大且可供性丰富的环境中也是如此，例如万维网的超媒体环境。在万维网上，示符的设计会考虑其预期的人类用户，从而抽象掉用户无需关心的底层交互细节（即超媒体控件），只呈现直观地引导用户行为所需的信息（即超媒体的信息部分）[2]。此外，示符的呈现会持续调整，以更好地适应利用可供性时动态变化的智能体—环境上下文。在万维网上，这是通过 HATEOAS 原则实现的。该原则可视为环境层面（服务器端）的支持，能够使可供性和示符的提供始终与当前应用状态（以及当前应用状态下预期的用户需求）保持一致。

2.2 超媒体客户端工程

超媒体客户端是能够进行超媒体驱动交互的应用程序，也就是说，它们利用可供性，并通过使用超媒体控件（例如链接和表单）来推进应用状态，而应用逻辑则与控件附带的信息相耦合，例如链接关系或表单操作和内容类型的标识符。随后，通过超媒体作为应用状态引擎，每次交互都会使客户端收到关于（新近）可用可供性的新信息，这些信息以超媒体的形式呈现。因此，

[1]例如， $Ability\text{-}B = \{< graspable, grasp >, < reachable, reach >, ...\}$，以及 $Affords\text{-}B = \{< ability\text{-}to\text{-}reach, item\text{-}is\text{-}reachable >, ...\}$。[2] 尽管可供性理论是根据功能层面来定义可供性的，但在人机交互领域，功能可供性通常会与认知可供性区分开来 [16]，以便将设计关注点分开，例如服务提供者与用户界面设计师之间的关注点。

超媒体客户端与特定的万维网资源和数据对象保持解耦，因此比专门针对特定 Web API 定制的客户端更能适应 API 变更 [1]。

如果由人类智能体做出有关动作的决策，那么超媒体客户端就会变得像如今的 Web 浏览器一样通用（且可复用）：客户端会反复执行以下步骤：1）解析并向人类智能体呈现示符（见第 2.1 节）；2）等待人类智能体基于显露的示能符进行推理并采取行动，以实现其目标；3）根据人类智能体的决策，使用超媒体控件推进应用状态。如果客户端本身有更高层次的目标需要实现，那么示符就必须携带明确无歧义的语义，以便机器能够可靠地处理它们。这可以通过<link relations 和操作类型的标准化标识符 来实现。例如，Hydra 核心词汇表 [20] 和超媒体控件本体（HCTL）[3] 支持对 RESTful Web API 进行语义描述，从而为研究如何为具有更复杂应用逻辑的可复用客户端建模示符奠定基础；随后，W3C 物联网（WoT） [18] 研究了如何使客户端应对越来越多连接到互联网的异构设备。为此，W3C WoT 在超媒体控件建模（复用 HCTL）的基础上更进一步，提供了一种易于使用的交互模式

1

用于定义事物描述（*TD*），即机器可读的规范，用以揭示设备（事物）可供性的高层交互语义。

尽管 WoT TD 能够实现客户端应用逻辑的复用，但也有研究致力于开发能支持更具适应性的超媒体客户端的交互规范。例如， [17] 中的客户端是可执行的关联数据规范，其中包含[4] N3 规则[4] ，用于以更具上下文感知的方式使用超媒体（例如，可以识别当前可利用的可供性），并且可以集成到分层任务网络工作流规范中，以呈现更复杂的行为 [2]。最后，RESTdesc 格式的交互规范 [32] 可作为 N3 推理器的输入，从而能够根据运行时系统目标动态组合可供性。

2.3 面向目标驱动交互的推理

自主智能体 能够展现适应性行为，因为它们处理的抽象使其能够推理自身的动作，并主动发起交互以实现其动态目标。例如，实现过程推理系统 [13] 的智能体能够发现并执行以计划形式表示的行为，并根据自身的信念、愿望和意图 [27]推理此类计划的适用性与相关性。此外，自主智能体还能通过自动规划方法 [14], , 根据期望效果推理动作的前置条件 和后置条件 ，从而合成新计划。

具体而言，与可供性利用相关的行为之所以成为可能，是因为上述抽象确立了智能体与其环境之间的关系。例如，Agents & Artifacts 元模型（A&A） [28] 将人工制品定义为组织在工作空间中的工具，智能体可通过这些工具感知并操控环境。随后，智能体能够置身于（逻辑）工作空间中，从而提高交互效率，并将其感知与动作的范围指向同处一地且感兴趣的人工制品。

为支持部署大规模且具有丰富可供性的基于 Web 的多智能体系统（MAS），人们也曾在万维网的背景下研究自主智能体及其针对目标和环境制品推理动作的能力。超媒体多智能体系统 [8] 是一类多智能体系统，其遵循万维网的架构原则，旨在赋予超媒体客户端应对不断演进的 Web API 的能力。超媒体多智能体系统中的智能体已经能够通过 WoT TD 发现并利用动态超媒体环境中的可供性 [8], ，但它们仍无法充分发挥推理和决策能力，因为 TD 并未专门用于建立与面向智能体编程和多智能体系统中的抽象（例如目标、动作前提条件等）的关联。

3 超媒体多智能体系统的示符

下文将介绍示符作为超媒体多智能体系统中的一等抽象，并在第 3.1 节中为此对示符进行形式化定义。示符的主要职责是支持可供性的可发现性和可解释性。因此，在第 3.2 节中，我们将介绍一种利用智能体—环境互补性的机制，以便只向身处超媒体环境中的智能体展示相关的示符。最后，在第 3.3 节中，我们将讨论如何定制示符，以支持具有不同能力的智能体，并介绍基于两种用于动作推理的知名方法——程序推理系统 [13] 和 STRIPS [12]——为智能体设计的示符。

3.1 面向智能体的超媒体驱动交互示符设计

本节将定义一种通用的示符模型，该模型 a）捕捉利用可供性所需的智能体—环境互补性；b）支持在万维网上以超媒体驱动的方式利用可供性。具体而言，该模型支持超媒体多智能体系统中自主智能体之间的交互。

3.1.1 示符的一般模型。 我们的示符模型建立在 Chemero 和 Turvey 于 [7] （见第 3.1 节）提出的可供性模型之上。因此，我们确定：

- 一种可供性 $aff_1 =< a_1,\ s_1 >$例如，机械臂的夹爪可闭合可供性；当自主智能体具备 a_1 通过设备的 HTTP API以操作员身份登录的能力，且情境为 s_1 夹爪处于打开状态 时，该可供性便可被利用（参见定义 1）；
- 一种能力 $a_0 =< aff_1,\ b_1 >$ 例如，当夹爪可闭合这一可供性可被利用时，智能体通过设备的 HTTP API 执行 b_1 闭合夹爪 这一行为的能力（参见定义 2）；
- 相关的可供性集合 $Affords\text{-}B$ （参见定义 3）和能力集合 $Ability\text{-}to\text{-}B$（参见定义 4），用于考察智能体如何根据行为集合 $B = \{b_1,\ b_2,\ \ldots\ldots\ b_n\}$利用系统中的可供性，例如，$B = \{gripper\text{-}closable,\ login,\ \ldots\ldots\}$。

[3]HCTL 本体可在线获取：https://www.w3.org/2019/wot/hypermedia [4]N3 规范可在线获取：https://www.w3.org/DesignIssues/Notation3

我们将示符 sig 定义为一种可感知的提示或标志，可通过有意义的解读揭示有关可供性 aff_1的信息，正式定义如下：

$$sig = (sp_b, A, c, salience), \tag{5}$$

其中：

- sp_b 是行为 b 的所指规范，属于集合 B；也就是说，这是当 aff_1 可被利用时，智能体能够执行的行动方案的规范；
- A 是一个能力集合，其中能力是智能体执行集合 B中某种行为的特质。因此， sig 建议智能体应具备集合 A 中的所有能力，以便aff_1 变得可被利用。
- c 是上下文， sig 建议持有该上下文，以便 aff_1 变得可被利用；也就是说，这是建议智能体—环境情境遵循的约束；
- $salience$ 是 sig 的一种特质，表明 sig 有多大用处或相关性。

RDF 中机器可理解的示符 [21] 根据列表 1（第 7–13 行）中的定义5构建：它表示闭合夹爪这一行为的规范，并建议在智能体能够成为操作员且夹爪处于打开状态时执行该行为。

请注意，示符（此处为 sig）的定义（及其表示）并不包含对可供性（此处为 aff_1）的任何直接引用。这样做有两个原因：首先，可供性是在<无样式>个体智能体</无样式>出现在恰当情境中时产生的，而示符则是针对<无样式>主体类型</无样式>定义的。这一点至关重要，因为设计者通常事先并不了解将会处于环境中的个体智能体。通过这种方式，定义5 保留了示符的可演化性和可复用性，因为智能体与其环境之间的耦合更加松散。其次，在可供性理论中，可供性被定义用于研究<无样式>动物</无样式>如何控制其行为。然而，我们对<无样式>示符</无样式>进行建模，是为了支持<无样式>自主智能体</无样式>之间的交互；与动物相比，这些智能体具有更加异质的认知能力和感觉运动能力——其中包括认知能力高度依赖计算过程（例如对表示进行推理）的智能体，以及其感觉运动能力[5] 能够实现远比动物行为更加广泛的行为。

尽管示符与可供性之间不存在正式的关系，示符的设计者 sig 仍可利用定义5中的要素，与 aff_1中的特定要素建立（直接或间接）关系。这种关系可以采取的形式可归纳为以下几种情况：

（C1） sp_b行为 b等同于对利用 aff_1 的行为 b_1 进行规格说明，以及对可能导致 aff_1 可被利用的任意行为进行规格说明，即 sp_b 规定集合 $B\prime \subseteq B$中智能体的行为，以及 $B\prime \supseteq \{b_1\}$；（C2） A 等同于导致 aff_1 可被利用的任意能力的集合，即 $A \subseteq Ability\text{-}to\text{-}B$；（C3） c 等同于在情境 s_1中得到满足的约束（的一部分）。

[5]在可供性理论中，动作被认为与感知严格耦合。尽管也有自主智能体展现出类似的认知能力和感觉运动能力，以执行从感知到动作的行为，例如 Rodney Brooks 的物理具身智能体，它们实现了包容架构， [5], 但超媒体多智能体系统预计将具有更多样化的智能体能力。

C1—C3 应根据环境设计者对目标智能体的经验和预期来使用，以促成有效且高效的交互。因此，设计者负责决定何时汇总或省略对集合 $Affords\text{-}B$ 和 $Ability\text{-}to\text{-}B$相关信息的引用。例如，（C1）允许管理行为 b 的粒度，该行为在所指规格 sp_b中得到规定。考虑到 aff_1是“夹爪可闭合”这一可供性，那么 b可以被规定为序列 $< b_2, b_1 >=< login, close\text{-}gripper >$。如果预期 b_2 和 b_1 会经常按顺序执行，或需要让智能体能够访问高层行为的规格，那么这种粒度可能是有益的。同样，（C2）使设计者能够避免在 A 中包含目标智能体预期始终具备的 $Ability\text{-}to\text{-}B$ 能力。（C3）通过确定哪些约束涉及使 aff_1 可被利用的情境 s_1 中最重要的方面，从而支持推荐上下文 c。通常，上下文可以包含有关智能体的约束，以及有关提供可供性的实体的约束；这些约束可能彼此无关（例如，对智能体意图的约束，以及分别针对制品状态的约束），也可能相互依赖（例如，智能体处于包含环境的同一工作环境中）。

在我们提出的示符模型中，推荐能力和上下文并非旨在约束智能体的行为，即限制智能体是否会实际利用某种可供性。不过，它们都有助于评估可供性利用的相关性，进而评估对于在特定情境下具有特定能力的智能体而言，该示符的相关性。

我们进一步完善模型，以定义如何为超媒体多智能体系统中的超媒体驱动交互指定行为，例如，采用 AgentSpeak 计划 [4], 、JADE 行为 [3], 等形式。鉴于这种多样性，我们对行为规范的定义仅施加一项形式化要求：至少指定一个动作。

通过以下方式指定动作：a) 表单，用于描述可实现和执行该动作的超媒体控件；以及 b) 可选的输入。例如，行为 close-

清单1：一个（通用）示符，揭示有关机械臂的夹爪可闭合可供性的信息。

```
1 @base <http:// ex.org/wksp/1/ arts/1>. 2 @prefix hctl:<https:// www .w3.org /2019/ wot/hypermedia #>. 3 ...7 <#sig> a hmas:示符 ; 8 hint:signifies <#close -gripper>; 9 hint:recommendsAbility [ a manu:操作能力 ]; 10 hint:recommendsContext <#env -context >. 11 12 <#env -context >a hint:上下文 ; sh: targetNode ex:leubot ; 13 sh:property [ sh:path manu: hasGripperValue ; 14 sh:hasValue "500"^^xsd:integer ] . 15 16 <#close -gripper> a hint:动作规范; 17 hint:hasForm [ hctl:hasTarget leubot:base ; 18 hctl:forContentType "application /json" ] ; 19 hint:expects [ a hint: 输入 ; 20 hint:hasSchema <#gripper-schema > ]. 21 22 <#gripper-schema >a js:ObjectSchema ; 23 js: properties [ a js:IntegerSchema ; 24 js: propertyName "manu:hasGripperValue "; js:enum "0"]].
```

夹爪可以通过一个表单指定为动作，该表单描述了一个 HTTP 请求以及一个夹爪输入值。动作执行可以被视为一种行为，预期会导致环境状态（例如夹爪状态）发生变化。在这种情况下，系统中任何监视智能体执行动作的实体，都可能感知到该智能体正在对环境采取行动。不过，在智能体执行动作以影响自身感知的情况下，动作也可用于感知。规范 sp_b 对这种行为（动作执行）进行了形式化描述 b，如下所示[6]：

$$sp_b = (Forms, \lfloor Input \rfloor), \tag{6}$$

其中，$Forms$ 是一组表单，每个表单都描述所指定动作的一种实现；$Input$ 是在动作执行可以或需要参数化时所需的输入。列表 1（第 15–24 行）呈现了闭合夹爪的所指规格。

3.1.2 自主智能体的能力。接下来，我们讨论示符所推荐的能力如何促进多智能体系统中的有效、高效可供性发现与利用。例如，基于示符进行推理的智能体可以推断，只有具备适当能力时，才应利用相关可供性。即使智能体当前不具备推荐的能力，仍可利用这些信息来扩展自身能力，或将目标委托给有能力的智能体。例如，如果智能体 ag_1 具备集合 A_1 中的能力，并且知晓示符 $sig_1 = (sp_b,\ A,\ c,\ salience)$，其中 $A \nsubseteq A_1$，那么 ag_1 可能会决定转发 sig_1，并请求具备集合 A_2中能力的智能体 ag_2 执行相关行为，其中 $A \subseteq A_2$。接下来，我们将考察智能体的能力，以说明示符设计者可以考虑的智能体多方面特征。

智能体可能具备基于其内部状态的表征进行推理的能力；例如，BDI智能体可以对其信念、愿望和意图进行推理 [27]。推荐这类能力可确保智能体具备执行指定行为所需的认知技能。例如，在交互协议的上下文中，若某个示符表示一项具有心理主义语义的发送消息规范（例如 KQML [19]），则它应推荐智能体具备心理状态的能力，以支持其遵循该协议。

此外，智能体还可能具备运用特定方法和机制对动作进行推理并提前规划的能力，例如，使用类STRIPS规划器来综合生成计划 [13]。因此，面向具备此类规划能力的智能体的示符，应表示有助于构建合适规划域的行为规范（例如，包含动作及其前置条件和后置条件的规范）。或者，面向实现程序推理系统 [13] 的BDI智能体的示符，也可以将行为规范表示为计划。

智能体还可以具备基于其社会情境采取行为的能力，例如，在工业车间操作机器，以履行机器操作员的角色。在多智能体环境（例如，组织 [33]）中推荐与角色相关的能力，可以帮助智能体根据角色判断哪些可供性可以利用，或帮助其履行某个角色。例如，一个指定某种在制造车间操作机械臂的行为，应推荐智能体具备扮演机器人操作员角色的能力，而不是仓库经理的角色。

智能体也可能具备在给定领域内运作的能力，例如，掌握工业制造中的抽象和流程知识。例如，使用特定语义模型来表示工业流程行为规范的示符，例如 SAREF4INMA[7] 语义模型，应当建议智能体具备解读所需模型的能力。

最后，智能体也可能具备在存在可供性时执行某种行为的能力，例如，根据行为规范采取行动。此类能力可能源自集合 $Ability\text{-}to\text{-}B$，示符对其的建议属于情况 (C2)。例如，表示闭合夹爪规范（列表 1）的示符可以建议智能体具备以操作员身份登录 的能力。

3.2 环境对示符暴露的支持

基于我们对示符的形式化定义，以及对所提出模型中可考虑的智能体能力多样类型的讨论，接下来我们将探讨如何在超媒体环境中实现示符的暴露并对其进行动态调整。

3.2.1 在环境中暴露示能符。示符并非简单的信息资源，而是在超媒体环境中可用的一种构造，使情境化智能体能够发现并解读可供性。这里，我们认为可以通过工作空间暴露示符；工作空间在逻辑上对环境进行划分（类似于 A&A 工作空间 [28]）[8]。

具体而言，超媒体环境中的工作空间是一个容器，包含 Web 资源以及这些 Web 资源之间发生的交互。工作空间 w的正式定义如下：

$$w = (R, I), \tag{7}$$

其中，R 是工作空间 w 中所含资源的集合（例如，智能体、示符、其他工作空间等），而 I 是发生在 w中的交互集合；这些交互由集合$IR \subseteq R$ 中的资源之间执行，这些资源提供并利用可供性。

我们进一步定义 $IR = Ag \cup Ar$，其中 Ag 是位于工作空间 w 中的智能体集合，而 Ar 是非自主实体的集合，即包含在 w中的人工制品 [28]。然后，设计示符以支持 IR中资源的交互，也就是说，支持 Ar 中的人工制品向 Ag中的智能体提供资源。为了考察环境在智能体间交互的上下文中提供了哪些可供性，智能体 $ag \in Ag$ 提供的可供性被

[6]我们插入 ⌊ 和 ⌋，以界定被视为可选的表达式。

[7]SAREF4INMA 为工业和制造领域定义了一套词汇表；可在线访问：https://saref.etsi.org/saref4inma/v1.1.2/。[8]我们将工作空间形式化，以便通过容器管理交互线索/元数据，为环境提供支持（例如，人机交互中的“自然映射”原理 [25]，或基于 Web 的系统中的 W3C 目录 [31] 和 API 生态系统 [24]）。不过，我们并不认为（也不强制要求）工作空间是管理多智能体系统环境和示符暴露的唯一方式。

视为身体[9] $body_{ag}$ ag $body_{ag} \in$ 的可供性，其中 Ar[10]。

然而，设计者的工作不应局限于构建定义5所规定的能指元组。相反，在为环境增添示符时，设计者还应考虑如何支持预期在环境运行时出现的智能体发现示符。[11] 为了宣传制品所提供可供性的示符，环境设计者可以创建一个制品档案，即通过示符以及通用（领域和应用特定）元数据来描述制品的结构化数据。形式上，制品的配置文件 p_{ar} 定义为以下结构：ar

$$p_{ar} = (Sig_{ar}, s_{ar}), p_{ar} \in R, \tag{8}$$

其中，Sig_{ar} 是配置文件 p_{ar} 中公开的示符集合，而 s_{ar} 是用于记录 ar 部分情境的元数据。

对于工作空间 w 中的制品 ar，w 中至少包含一个制品档案 p_{ar}，以明确记录 ar 与 w 之间的包含关系。这样，便可通过 w 包含 ar，实现对 p_{ar} 中示符集合 Sig_{ar} 的发现。

3.2.2 动态调整示符暴露。在大型环境中，随着人工制品、可供性以及相应示符数量的增加，需要为示符的可发现性提供额外支持。同时，在开放环境中，智能体基于各自不同的能力来解读示符，这可能会妨碍示符的可解释性。

为解决这些问题，我们提出一种示符暴露机制(SEM)，根据目标智能体的能力和情境（例如智能体的目标），以及人工制品的情境（例如根据当前人工制品状态下的有效转换），暴露经过筛选的一组示符。

首先，我们考虑集合 P_{Ag}，该集合包含工作空间 w 中每个智能体 $ag \in Ag$ 的配置文件。然后，每个配置文件 $p_{ag} \in P_{Ag}$ 都是描述智能体 ag 能力和情境的元数据。智能体 ag 的配置文件 p_{ag} 正式定义如下：

$$p_{ag} = (A_{ag}, s_{ag}), p_{ar} \in P_{Ag} \subset R, \tag{9}$$

其中，A_{ag} 是智能体 ag 的能力集合，s_{ag} 是捕获智能体 ag 部分情境的元数据。

此外，我们考虑集合 P_{Ar}，该集合包含工作空间 w 中每个制品 $ar \in Ar$ 的配置文件（参见定义 8）。

现在，我们给出 SEM 的定义（参见 [22]），其功能描述如下：给定一个智能体档案 $p_{ag} \in P_{Ag}$ 和一个制品档案 $p_{ar} \in P_{Ar}$，SEM 会输出一个制品档案 p_{ar}，其中仅公开满足以下条件的 p_{ar} 的示符：a）与智能体 ag 的能力相匹配；b）与智能体 — 环境情境相匹配（即 ag-ar 情境）。对于前者，我们只需判断推荐能力集合是否为该智能体能力集合的子集。对于后者，我们使用评估函数 E，评估智能体的情境和制品的情境在多大程度上符合示符所推荐的上下文。为此，我们将 C_{ar} 视为可通过制品 ar 的配置文件访问的所有推荐上下文的集合。然后，我们用 SC_{Ar} 表示可通过制品档案集合 P_{Ar} 访问的所有推荐上下文的集合，并将其与相应的 P_{Ar} 配置文件中记录的情境配对。配对 $(c_{ar},\ s_{ar}) \in SC_{Ar}$ 用于将制品 ar 的情境仅与适用于同一制品 ar 的上下文关联起来：

$$SC_{Ar} = \{C_{ar} \times \{s_{ar}\} \mid p_{ar} = (Sig_{ar}, s_{ar}) \wedge ar \in Ar\}. \tag{10}$$

我们还用 S_{Ag} 表示一个集合，其中包含可通过智能体档案集合 P_{Ag} 访问的情境，形式化定义如下：

$$S_{Ag} = \{s_{ag} \mid p_{ag} = (A_{ag}, s_{ag}) \wedge ag \in Ag\} \tag{11}$$

接着，我们定义评估函数 E，用于针对集合 CS_{Ar} 中的上下文，验证集合 S_{Ag} 中的智能体情境和集合 CS_{Ar} 中的制品情境：

$$E : CS_{Ar} \times S_{Ag} \longrightarrow [0, 1]. \tag{12}$$

最后，$P_{\prime}Ar$ 是可能生成的制品档案集合：

$$P'_{Ar} = \{(Sig'_{ar}, s_{ar}) \mid p_{ar} = (Sig_{ar}, s_{ar}) \wedge Sig'_{ar} \in \mathcal{P}(Sig_{ar}) \ \wedge \ ar \in Ar\} \tag{13}$$

现在，我们可以将 SEM 形式化定义如下：

$$\begin{aligned} SEM : P_{Ag} \times P_{Ar} \times \{e | e \in [0, 1]\} \longrightarrow P'_{Ar}; \\ ((Sig_{ar}, s_{ar}), (A_{ag}, s_{ag}), t) \longmapsto (\{sig = (sp_b, A, c, salience) \in Sig_{ar} \\ \mid A \subseteq A_{ag} \wedge E((c, s_{ar}), s_{ag}) > t\}, s_{ar}) \end{aligned} \tag{14}$$

其中，t 是作为输入提供给 SEM 的阈值。

虽然公开的指示符始终关联到被认为可利用或相关的可供性可能有益，但这并非必要。例如，在某些情况下，关于智能体和人工制品的观察不足以评估相关性，比如在高度动态的环境中，或者智能体不愿分享此类信息时。此外，具备规划能力的智能体可能需要一个与当前未必可利用的可供性相关联的动作空间。

3.3 针对智能体异构推理能力定制指示符

本节将探讨如何扩展通用指示符，以揭示与具备不同推理和规划能力的智能体相关的信息。具体而言，我们将考虑以下指示符的设计：a）实现过程推理系统（PRS）的 BDI 智能体；b）能够使用类 STRIPS 规划器规划自身行动的智能体。

为说明我们的方法，我们使用列表 1 中的通用指示符，它表示用于闭合机械臂夹爪的动作规范。下文将展示如何扩展此指示符，以定制其内容，使之适应目标智能体的能力。

[9]智能体的身体是一种制品，使智能体能够处于工作空间中，并与工作空间中的其他资源进行交互 [29]。[10]根据可供性的定义（参见定义 1），我们只考虑智能体与环境（智能体与制品）之间的交互。智能体间交互通过智能体身体归入智能体与环境交互，而环境与环境（制品与制品）之间的交互不予考虑。[11]相较于人机交互文献，这显著拓宽了示符概念的范围；在该文献中，人们认为示符只有在边缘情况下才能在运行时进行调节。

3.3.1 定制用于基于*PRS*的推理的指示符。具有基于PRS的能力的BDI智能体需要寻找指示符，以满足其意图，即该智能体承诺实现的目标。尽管该智能体的计划库中可能已经包含一个计划，也就是实现其目标的行动方案，但动态环境并不总能允许将所需动作与特定的动作实现相匹配。更具体地说，如果智能体能够使用抓取与放置计划，它可能打算操作机械臂（例如，闭合机械臂的夹爪），但却不知道在当前工作空间中该使用哪台机械臂，也不了解其他更低层级的交互细节。

为了在我们的方法中发现合适的指示符，智能体可以更新其配置文件，以描述其心理态度（部分）——例如当前的信念和目标，包括其在寻找满足愿望的适当手段时实现特定目标的愿望。考虑到某智能体希望实现拾取物品并将其放置在指定位置这一目标，该智能体的配置文件可能与列表 212 中所示的配置文件类似。

考虑到基于 PRS 的智能体的能力，设计者现在可以构建指示符，推荐与目标智能体预期情境相关的上下文，即以信念、愿望和意图表示的情境。列表 3 展示了一个关于机械臂的夹爪可闭合可供性的指示符示例；该指示符扩展了同一可供性的通用指示符（列表 1），以更好地适配此类智能体。在本例中，所推荐的上下文是一组约束，智能体—环境情境必须对其进行验证，指示符才会被视为与该智能体相关。这些约束以形状约束语言（SHACL）[13] 表示，涉及智能体—人工制品情境的以下方面：1）智能体希望拾取一件物品并将其放置在目标位置（第 20–43 行）；2）物品处于提供夹爪可闭合可供性的机械臂的作用范围内（第 45–59、61–65 行）；3）目标位置处于提供夹爪可闭合可供性的机械臂的作用范围内（第 45–59、61–65 行）。

此外，该指示符还会推荐一种基于*PRS*的能力。因此，如果具备此类能力的智能体在运行时查找指示符，且有可用的 SEM（见第 3.2 节），SEM 将公开列表 3 中的定制化 指示符，而非列表 1 中的通用指示符。最后，如果 SEM 具备 SHACL 验证功能，则会根据智能体的情境（列表 2）是否符合 SHACL上下文形状[14]来调整公开内容。在这种情况下，仅当智能体—环境情境符合推荐约束时，才会向具备基于PRS的能力的智能体公开定制化指示符。

[12]我们认为，该智能体具有一种愿望，即实现 AgentSpeak语言 [4] 中指定的目标。[13]SHACL 规范可在线查阅：https://www.w3.org/TR/shacl/

[14]此前已在 [30]中介绍过带有 SHACL处理器的 SEM。在那里，智能体会指定对指示符施加约束的 SHACL形状，并在查找符合条件的指示符时，将这些形状作为输入提供给 SEM。另一方面，本文考虑由环境设计者创建指示符，并指定用于在运行时识别符合条件的智能体—环境情境的约束。

清单 2：一个实现 PRS、并希望抓取与放置物品的 BDI智能体的资源描述文件。

```
1 @base <http:// ex.org/wksp/1/ arts/2>。
2 < 1 3 >
3
@prefix item -profile: http:// ex.org/wksp/ / arts/ # ...11 <>a
hmas:AgentProfile ; hmas:isProfileOf <#agent> 。1213 <#agent>a
hmas:Agent ; 14 hint:hasAbility [a prs:PRS能力 ] ; 15 hint:hasAbility [
a manu:操作能力 ] ; 16 prs: hasDesire [ a prs:目标达成 ,manu: 抓取与放
置 ; 17 prs: hasInputList [ a rdf:List ; 18 rdf:first wksp:item ; 19 rdf:rest
[ a rdf:List ; 20 rdf:first <#位置>] ] 。21 22 item -profile :item a
manu:物品 ; 23 manu:hasLocation item -profile :location 。
```

清单 3：为基于 PRS 实现 BDI架构的智能体定制的指示符。

```
1 ...9 <#sig> a hmas:Signifier ; 10 hint:signifies <#close -gripper> ;
11 hint: recommendsAbility [ a prs:PRS能力 ] ; 12 hint:
recommendsAbility [ manu:操作能力 ]; 13 hint: recommendsContext <
#env -context >, <#prs -context >. 1415 <#prs -context >a hint:Context ;
sh: targetClass hmas:Agent ; 16 sh: property [ sh:path prs:hasDesire ; 17
sh:minCount 1 ; sh: qualifiedMinCount 1 ; 18 sh: qualifiedValueShape <#
desire -shape> ]. 1920 <#desire -shape> a sh:节点形状 ; 21 sh:class
manu:抓取与放置 ; 22 sh: property [ sh:path prs:hasInputList 43 ... ]. 44
45 <#item -shape> a sh:节点形状 ; 46 sh:class manu:物品 ; 47 sh: property [
sh:path manu:hasLocation ; 59 ... ]. 6061 <#location -shape>a sh:节点形状 ;
62 sh:class manu: 位置 ; 63 sh: property [ sh:path manu: 范围内 ; 64
sh:minCount 1 ; 65 sh:hasValue ex:leubot ].
```

列表 4：具备 STRIPS 规划能力的智能体的定制化指示符。

```
1 @prefix pddl: 2 <http:// www.cs.yale.edu /homes/dvm /
daml/pddlonto .daml#>. 3 ...9 <#sig> a hmas:示符 ; 10 hint:signifies <#
close -gripper> ;11 [ hint: recommends能
力 a strips:STRIPS 12 规划能力 ]. 1314 <#close -gripper> a hint:动作规范;
15 ... 21 a a pddl:动作 ; 22 pddl:action -label "closeGripper "; 23 pddl:参数 [
a pddl:参数_seq ; 24 rdf: _1 <#参数1> ];25 [
pddl:前置条件 ... ;32 [ 31 pddl:效果 ... ]. 4243 <#参数1> a
pddl:参数 ; 44 pddl:名称 "?gv" 41 ] ; 45 drs:type manu:夹爪值 ; 46 :hasSche
ma <#gripper-schema >.
```

3.3.2 面向基于 *STRIPS* 的推理定制指示符。与利用那些以智能体的信念、愿望和意图为设计考量的指示符不同，具备自动规划能力的智能体需要能够指示动作规范的指示符，以便丰富规划域。例如，列表 4 展示了机械臂夹爪可闭合这一可供性的定制化指示符。它扩展了列表 1 中的通用指示符，使其能够指示与具备 STRIPS 规划能力的智能体相关的动作规范。该动作规范指定了可执行动作的类型，并进一步规定了执行该动作的前置条件和效果。在这个示例中，我们重新使用了 [23] 中介绍的 PDDL 本体来指定可用于丰富 PDDL 域的动作，从而使其成为适用于 PDDL 自动规划器的输入。该指示符还扩展为推荐 *STRIPS* 规划能力，使 SEM 能够根据请求智能体的能力调整指示符暴露。

4 实现与部署

为了将我们的方法应用于具体用例，我们介绍一个原型超媒体多智能体系统及其演示场景。

4.1 原型部署

在我们的实现 15 中，我们使用 Yggdrasil 平台作为超媒体多智能体系统 [8] 的存储库，用于存放超媒体环境。我们扩展了 Yggdrasil API，以支持发布第 3.1 节所述的智能体档案和制品档案。因此，该 API 支持基于以下内容发布和管理资源档案：a) 超媒体多智能体系统核心本体（HMAS-core），用于描述超媒体多智能体系统的核心方面（例如，工作空间、资源档案等）；以及 b) 超媒体多智能体系统交互本体，该本体基于我们的示符模型扩展了 HMAS-core。

我们还为 Yggdrasil 扩展了一个 SEM，使其能够根据第 3.2 节所述的形式模型动态调整示符暴露。具体而言，智能体档案和制品档案可以发布为可通过超媒体环境中的工作空间访问的形式。因此，当智能体在超媒体环境中发现制品档案时，SEM 负责评估应向该智能体暴露哪些示符：SEM 会识别当前正在寻找示符的智能体，并尝试获取该智能体的档案。如果没有可用的智能体档案，则会将制品档案提供给智能体，而不对示符进行任何调整。另一方面，如果 SEM 能够访问智能体档案，则会依据定义 14 评估该智能体与制品之间的互补性。例如，如果智能体具有特定能力，SEM 就会构建制品档案的一个变体，仅暴露包含针对这些特定智能体能力的建议的示符。

在原型中，我们使用了 Jason [4] 智能体，它们是在 JaCaMo[16] 应用程序中实现并部署的。所用的指示符旨在揭示 PhantomX AX12 Reactor 机械臂[17]可供性的信息。

4.2 部署场景

为了验证原型实现，我们考虑了这样一种超媒体多智能体系统：在一个配备工业设备的工作空间中，按照以下场景进行可供性发现与利用：某制造工作空间中有一台机械臂，它为位于该工作空间中的智能体提供可供性，例如移动夹爪和机械臂底座的可供性。机械臂被建模为一个制品，并通过包含在工作空间中的制品配置文件 在超媒体环境中标示其存在。机械臂可供性相关信息由其配置文件中公开的示符揭示；设计这些示符时考虑了预期位于制造工作空间中的智能体具备的不同能力。具体而言，我们针对智能体的能力考虑了三种类型的示符：

- 为实现程序推理系统（PRS）的 BDI智能体推荐能力的指示符。
- 为能够使用类STRIPS规划器进行自动规划的智能体推荐能力的指示符。
- 未针对具备任何特定动作推理与执行能力的智能体进行定制的通用指示符。

一个实现 PRS 的 BDI智能体加入工作空间，目标是抓取并放置一件物品。超媒体环境通过该智能体的配置文件显示其存在于工作空间中；该配置文件包含以下元数据：a）智能体的身体位于工作空间内；b）智能体希望执行抓取与放置；c）智能体具备 PRS 能力。该智能体已通过计划库获取了相关的抓取与放置计划，但并不了解有效执行该计划所需的任何底层实现细节。为此，该智能体会查找指示符，以获取有关可用于利用相关可供性的超媒体控件的信息。该智能体查找制造工作空间内机械臂的可供性信息时，会从 SEM 获得与自身能力相关的指示符。

另一名智能体带着同样的目标加入工作空间：抓取并放置一件物品。尽管该智能体当前没有相关的抓取与放置计划，但它能够使用类似 STRIPS 的规划器执行自动规划——这一能力在该智能体的配置文件中有所体现。该智能体决定寻找可能有助于合成抓取与放置计划的示符。将注意力集中到机械臂后，该智能体会收到机械臂的配置文件，其中仅暴露了与该智能体的 STRIPS 规划能力相关的示符。因此，该智能体得以访问动作规范；这些规范说明了可对机械臂执行的动作类型，以及执行这些动作的前提条件和效果，适合用于丰富规划域。

[15]实现可在线获取：https://github.com/Interactions-HSG/yggdrasil/tree/feature/sem [16]JaCaMo 文档可在线获取：https://jacamo.sourceforge.net/

[17]https://www.trossenrobotics.com/p/phantomx-ax-12-reactor-robot-arm.aspx

5 结论

本文将示符引入为超媒体多智能体系统中的一等抽象。为此，我们对示符设计进行了形式化，使其能够通过超媒体驱动的方式利用万维网上的可供性，并刻画利用可供性所需的智能体—环境互补性。这些示符与一种机制相结合，该机制描述了如何根据智能体—环境情境动态调整示符暴露。我们通过示例说明，示符如何借助超媒体将行为与其实现解耦，并可针对智能体不同的推理能力进行定制，从而在可供性丰富且开放的、基于万维网的多智能体系统中，实现有效且高效的可供性发现与利用。基于本文提出的工作，我们旨在依据 HATEOAS 原则以及可供性理论中的行为—能力非良基性，通过设计和暴露示符，支持并评估更复杂行为的有效性和效率。

致谢

本研究获得了瑞士国家科学基金会第189474号资助（*HyperAgents*），以及欧盟地平线2020研究与创新计划第957218号资助（*IntellIoT*）。我们感谢 Yousouf Taghzouti 对语义内容协商主题所提供的宝贵意见。

参考文献

[1] Mike Amundsen。2017年。*RESTful Web* 客户端：通过超媒体实现复用。O’Reilly Media, Inc。[2] Nico Aßfalg、Helen Schneider 和 Tobias Käfer。2021年。可读写关联数据上的集成规划与执行。载于全体智能体挑战赛（*ATAC*）第*20*届国际语义网会议（*ISWC*）论文集。25–31。[3] Fabio Luigi Bellifemine、Giovanni Caire 和 Dominic Greenwood。2007年。使用 *JADE* 开发多智能体系统。John Wiley & Sons。[4] Rafael H Bordini、Jomi Fred Hübner 和 Michael Wooldridge。2007年。使用 *Jason* 以 *AgentSpeak* 编程多智能体系统。John Wiley & Sons。[5] Rodney Allen Brooks。1999年。寒武纪智能：新人工智能的早期历史。麻省理工学院出版社。[6] Victor Charpenay、Sebastian Käbisch 和 Harald Kosch。2016年。引入事物描述与交互：物联网本体。载于与第*15*届国际语义网会议（*ISWC*）联合举办的第*3*届流推理（*SR*）和第*1*届物联网语义网技术（*SWIT*）研讨会联合论文集。55–66。[7] Anthony Chemero 和 Michael Turvey。2007年。复杂性、超集与感知—行动的生态学视角。*Biological Theory* 2，1（2007），23–36。[8] Andrei Ciortea、Olivier Boissier 和 Alessandro Ricci。2018年。使用超媒体构建万维网多智能体系统。载于*Weyns, D.*、*Mascardi, V.*、*Ricci, A.*（编）多智能体系统工程（*EMAS*）（计算机科学讲义）。285–301。[9] Andrei Ciortea、Simon Mayer、Fabien Gandon、Olivier Boissier、Alessandro Ricci 和 Antoine Zimmermann。2019年。回顾十年：多智能体系统与万维网之间缺失的桥梁。载于第*18*届自主智能体与多智能体系统国际会议（*AAMAS*）论文集。[10] Roy T. Fielding。2000年。基于网络的软件架构的架构风格与设计。博士学位论文。加州大学欧文分校。[11] Roy T. Fielding。2008年。REST API 必须由超文本驱动。检索日期：2022年10月28日，网址：https://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven [12] Richard E. Fikes 和 Nils J. Nilsson。1971年。STRIPS：将定理证明应用于问题求解的新方法。人工智能 2，3–4（1971），189–208。[13] Michael P. Georgeff 和 Amy L. Lansky。1987年。反应式推理与规划。载于第*6*届全国人工智能会议（*AAAI*）论文集，第87卷。677–682。

[14] Malik Ghallab、Dana Nau 和 Paolo Traverso。2016。自动规划与行动。剑桥大学出版社。[15] James J. Gibson。2014。视觉感知的生态学方法：经典版。心理学出版社。[16] Rex Hartson。2003。交互设计中的认知、物理、感官和功能可供性。行为与信息技术22，5（2003），315–338。[17] Tobias Käfer 和 Andreas Harth。2018。面向关联数据的用户代理基于规则的编程。载于与 *2018* 年 *Web* 大会（*WWW*）同期举办的万维网上的关联数据研讨会（*LDOW*）。[18] Matthias Kovatsch、Michael McCool、Sebastian Käbisch、Victor Charpenay 和 Takuki Kamiya。2020。物联网（*WoT*）事物描述。W3C 推荐标准。W3C。https://www.w3.org/TR/2020/REC-wot-thing-description-20200409/。[19] Yannis Labrou 和 Tim Finin。1997。新 *KQML* 规范提案。技术报告。技术报告 TR-CS-97-03，马里兰大学。[20] Markus Lanthaler 和 Christian Gütl。2013。Hydra：一种用于超媒体驱动的 Web API 的词汇表。载于与 *2013* 年 *Web* 大会（*WWW*）同期举办的第八届万维网上的关联数据研讨会（*LDOW*）论文集。[21] Markus Lanthaler、David Wood 和 Richard Cyganiak。2014。*RDF 1.1* 概念与抽象语法。W3C 推荐标准。W3C。https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/。[22] Jérémy Lemée、Danai Vachtsevanou、Simon Mayer 和 Andrei Ciortea。2022。面向可供性驱动的多智能体系统的示符。在第 21 届自主智能体与多智能体系统国际会议（AAMAS）上举行的多智能体系统工程国际研讨会（EMAS）宣读的论文。[23] Drew McDermott 和 Dejing Dou。2002。在 RDF 中表示析取和量词。载于 *Horrocks, I., Hendler, J.*（编）《语义网（*ISWC*）*2002*》（计算机科学讲义）。Springer，250–263。[24] Mehdi Medjaoui、Erik Wilde、Ronnie Mitra 和 Mike Amundsen。2021。持续 *API* 管理。“O’Reilly Media, Inc.”。[25] Don Norman。2013。日常事物的设计：修订与扩充版。Basic Books。[26] Donald A Norman。2008。我眼中的 IT：示能 signifier，而非可供性。交互（2008）。[27] Anand S. Rao、Michael P. Georgeff 等。1995。BDI 智能体：从理论到实践。。载于第一届多智能体系统国际会议论文集，第 95 卷。312–319。[28] Alessandro Ricci、Michele Piunti 和 Mirko Viroli。2011。多智能体系统中的环境编程：基于制品的视角。自主智能体与多智能体系统23，2（2011），158–192。[29] Alessandro Ricci、Mirko Viroli 和 Andrea Omicini。2006。CArtAgO：用于基于制品的环境原型开发的框架。载于*Weyns, D., Parunak, H.V.D., Michel, F.*（编）《多智能体系统环境 *III*（*E4MAS*）*2006*》（计算机科学讲义）。Springer，67–86。[30] Yousouf Taghzouti、Danai Vachtsevanou、Simon Mayer 和 Andrei Ciortea。2022。迈向语义内容协商。载于第 *23* 届知识工程与知识管理国际会议（*EKAW*）。[31] Farshid Tavakolizadeh、Andrea Cimmino、Michael McCool 和 Kunihiko Toumura。2023。物联网（*WoT*）发现。候选推荐标准。W3C。https://www.w3.org/TR/2023/CR-wot-discovery-20230119/。[32] Ruben Verborgh、Vincent Haerinck、Thomas Steiner、Davy Van Deursen、Sofie Van Hoecke、Jos De Roo、Rik Van de Walle 和 Joaquim Gabarro。2012。传感器 Web API 的功能组合。载于第五届语义传感器网络国际会议（*SSN*）论文集。65–80。[33] Gerhard Weiss。2000。多智能体系统，第二版。麻省理工学院出版社。

此图“sem-2.png”可通过以下地址获取 PNG 格式版本：

http://arxiv.org/ps/2302.06970v1

该图“aamas2023logo.png”可从以下位置获取，格式为“PNG”：

http://arXiv.org/ps/2302.06970v1

该图“sem.png”可从以下位置获取，格式为“PNG”：

http://arxiv.org/ps/2302.06970v1