Interaction Protocols in an Imperative Agent-Oriented Programming Language: the case of BSPL and SARL

Extended Abstract

Matteo Baldoni
Università degli Studi di Torino,
Dipartimento di Informatica
Torino, Italy
matteo.baldoni@unito.it

Cristina Baroglio
Università degli Studi di Torino,
Dipartimento di Informatica
Torino, Italy
cristina.baroglio@unito.it

Stéphane Galland
Université de Technologie de Belfort
Montbéliard, UTBM, CIAD UR 7533
Belfort, France
stephane.galland@utbm.fr

Roberto Micalizio
Università degli Studi di Torino,
Dipartimento di Informatica
Torino, Italy
roberto.micalizio@unito.it

Fatma Outay
College of Technological Innovation,
Zayed University
Dubai, United Arab Emirates
fatma.outay@zu.ac.ae

Stefano Tedeschi
Università della Valle d'Aosta
Université de la Vallée d'Aoste
Aosta, Italy
s.tedeschi@univda.it

ABSTRACT

This paper presents a framework for implementing BSPL interaction protocols in the SARL agent-oriented programming language. BSPL defines interaction protocols based on information flow, while SARL provides tools for building autonomous distributed systems. We introduce a set of transformation rules to automatically generate adapters, which allow SARL agents to enact BSPL-specified protocols with minimal manual effort required to agent developers. By aligning BSPL's declarative nature with SARL's agent behaviors and event-driven architecture, this approach facilitates the development of flexible, interaction-driven multi-agent systems.

KEYWORDS

Interaction protocols; Engineering MAS; SARL; BSPL

ACM Reference Format:
Matteo Baldoni, Cristina Baroglio, Stéphane Galland, Roberto Micalizio, Fatma Outay, and Stefano Tedeschi. 2025. Interaction Protocols in an Imperative Agent-Oriented Programming Language: the case of BSPL and SARL: Extended Abstract. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 2 pages.

1 INTRODUCTION

SARL [2] is a multi-threaded, imperative, agent-oriented language and platform designed for creating autonomous, decentralized, and interactive multi-agent systems (MAS). SARL's architecture is flexible and extensible, providing abstractions that emphasize agent autonomy, distribution, and concurrency. SARL, along with other approaches like ASTRA and LightJason, aims to bridge the gap with mainstream programming languages, focusing on an imperative approach unlike other proposals such as AgentSpeak(L), Jason, or GOAL. SARL supports various communication channels through the concept of *spaces*, which can be customized to fit different interaction needs. The SARL metamodel is based on the key concepts of *agent*, *behavior*, *capacity*, and *skill*. However, SARL lacks specific constructs for implementing interaction protocols, requiring developers to hard-code interaction steps within the agents' logic, which impacts modularity and ease of use. Interaction protocols are crucial in distributed computing and MAS for effective communication and coordination among agents. BSPL [3] is a declarative specification that captures data dependencies in messages without explicitly ordering them, contrasting with procedural approaches like FIPA-ACL and AUML. BSPL protocols define roles, parameters, and messages, with parameters categorized as "in" or "out" to indicate data flow. Messages are sent based on the availability of required information, decoupling message emission from strict sequencing. The integration of BSPL into SARL enables the automatic generation of protocol *adapters* from BSPL specifications that could be used by SARL agents without extra coding effort.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

2 IMPLEMENTING PROTOCOLS IN SARL

Our methodology for integrating BSPL with SARL is based on the transformation of BSPL's declarative protocol specifications into SARL's imperative programming constructs. BSPL roles are translated into SARL as `ProtocolRole`, which defines abstract entities responsible for participating in protocols. Additionally, a `ProtocolSpace` is established in SARL to serve as a communication medium, ensuring isolated and efficient interaction among agents enacting a protocol. This approach leverages SARL's event-driven architecture to handle asynchronous communication, where BSPL messages are mapped to SARL events, with message parameters encoded as event attributes. To support protocol-related tasks, agents are equipped with `ProtocolCapacity` and corresponding `ProtocolSkill`, enabling them to compute enabled messages and send them as required. The `ProtocolReactiveBehavior` allows agents to handle incoming messages by updating their local state with received information, ensuring compliance with BSPL specifications. The enactment process in SARL involves agents dynamically creating protocol spaces and inviting other agents to participate, thereby ensuring privacy and efficiency in communication.

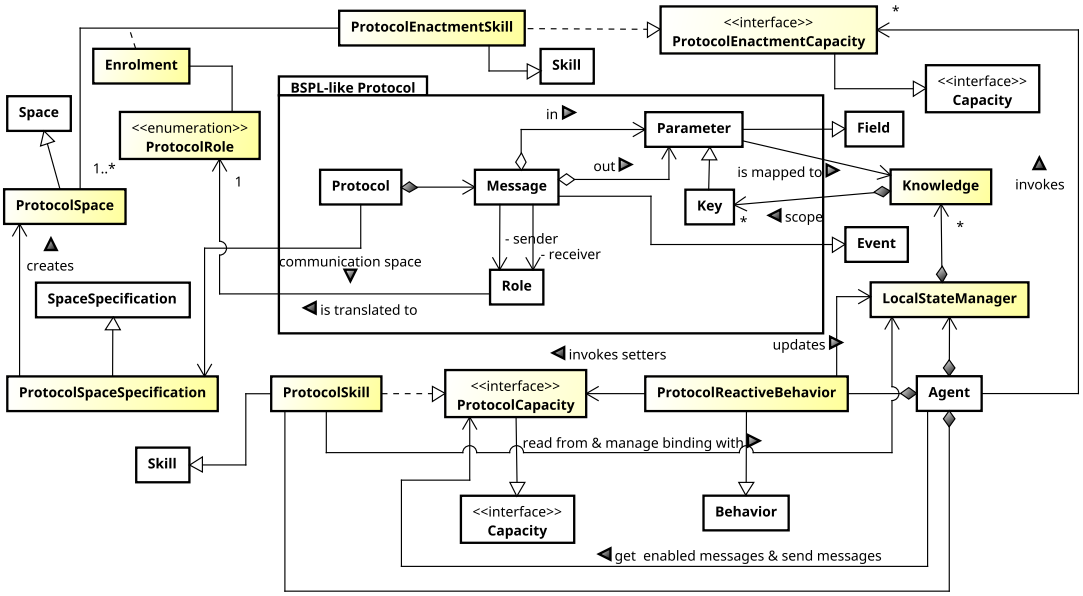

Figure 1: Mapping of the key BSPL and SARL concepts. Yellow indicates those new concepts introduced to support the mapping.

Algorithms and behaviors are defined to manage the sending and receiving of messages, maintaining the integrity of the protocol enactment. Figure 1 illustrates the conceptual mapping between the BSPL and SARL metamodels.

Additionally, we formalize a set of transformation rules that are necessary to produce protocol adapters for agents. For example, Equation 1[1] describes the transformation for generating the message receiving behavior of the agent, where $\alpha \xrightarrow{\beta} \gamma$ is valid when α was transformed for obtaining γ, assuming $\gamma \in \beta$, $\sigma[LS]$ is the agent local state, $\sigma[B_r]$ is set of agent's event handlers, b_m is the is run when an event receiving E.

$$\frac{\begin{array}{ccc} m \in \mathbb{M} & m \xrightarrow{\sigma[E]} e & r = rcvr(m) \\ b_m = (e) \mapsto \sigma[LS/LS \cup p_e] & & \sigma' = \sigma[B_r/B_r \cup \{b_m\}] \end{array}}{\langle m, \sigma \rangle \rightarrow \langle b_m, \sigma' \rangle} \quad (1)$$

Consider the following message, taken from the BSPL specification of the Contract Net Protocol in [1].

```
Participant -> Contractor : Propose
[in IDt key, in IDp key, in task, out offer, out pdecision]
```

The generated SARL code[2] is:

```
behavior ContractorReactiveBehavior {
    on Propose {
        var keys = #[occurrence.IDt, occurrence.IDp]
        #[keys, "task"].knowledge = occurrence.task
        #[keys, "offer"].knowledge = occurrence.offer
        #[keys, "pdecision"].knowledge = occurrence.pdecision
    }
}
```

[1]Bottom part is the transformation source and destination; Top part provides the conditions to apply.
[2]Available at https://github.com/sarl/sarl-protocols.

3 CONCLUSION

The primary contribution of this work is the development of a conceptual and technical framework that enables the automatic generation of SARL-compatible adapters from BSPL protocol specifications. This framework facilitates the seamless enactment of interaction protocols within SARL agents, leveraging the declarative nature of BSPL and the imperative, event-driven architecture of SARL. This approach bridges the gap between declarative protocol design and practical agent-oriented implementations, allowing developers to harness the strengths of both paradigms. Adopting a purely reactive approach to protocol enactment and defining different models of local state management within agents could improve scalability and adaptability in various multi-agent system applications. These advancements aim to refine the framework, making it more intuitive and efficient for large-scale implementations.

ACKNOWLEDGMENTS

Funded by the European Union – NextGenerationEU, Mission 4 Component 1.5 – ECS00000036 – CUP B63B22000010001 (C. Baroglio and S. Tedeschi). Funded by the Research Cluster R19098 of Zayed University, UAE (S. Galland and F. Outay).

REFERENCES

[1] Matteo Baldoni, Samuel H. Christie V, Amit Chopra, and Munindar P. Singh. 2024. Jason+BSPL: Including Communication Protocols in Jason. In *Agent Toolkits Special Track of the 21st European Conference on Multi-Agent Systems (EUMAS)*. Dublin, Ireland.

[2] Sebastian Rodriguez, Nicolas Gaud, and Stéphane Galland. 2014. SARL: A General-Purpose Agent-Oriented Programming Language. In *The 2014 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, Vol. 3. IEEE Computer Society Press, Warsaw, Poland, 103–110. https://doi.org/10.1109/WI-IAT.2014.156

[3] Munindar P. Singh. 2011. Information-driven interaction-oriented programming: BSPL, the blindingly simple protocol language. In *10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan,*, Vol. 1–3. 491–498.