Dynamic Service Composition Orchestrated by Cognitive Agents in Mobile & Pervasive Computing

Oscar J. Romero
Machine Learning Department
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, USA

***Abstract*—Automatic service composition in mobile and pervasive computing faces many challenges due to the complex nature of the environment. Common approaches address service composition from optimization perspectives which are not feasible in practice due to the intractability of the problem, limited computational resources of smart devices, service host's mobility, and time constraints. Our main contribution is the development of a cognitively-inspired agent-based service composition model focused on bounded rationality rather than optimality, which allows the system to compensate for limited resources by selectively filtering out continuous streams of data. The evaluation of our approach shows promising results when compared against state-of-the-art service composition models.**

***Index Terms*—Service composition, Middleware, Mobile and Pervasive Computing, Artificial Cognition**

I. Introduction

Service composition refers to the technique of creating composite services by the aggregation of atomic services. Despite the existence of Mobile/Pervasive Computing (MPC) middleware for service composition [4], [10], [11], there are still some challenges that need to be tackled. Thus, we claim that service composition should: (1) consider requests from multiple users; (2) consider resource scarcity in smart devices; (3) perform dynamic adaptation to unpredictable changes; and (4) deal with both short and long-term user's goals. Performing service composition while taking into account the myriad of variable factors mentioned above makes the problem become intractable even for approaches that use dynamic composition and focus on optimality-based (e.g, graph-, rule-, and workflow-based) solutions, which do no consider limitations imposed by the decision-making process (specially on smart devices). Thus, we propose a cognitively-inspired, bounded-rationality-based approach so-called *COPERNIC*, which seeks satisfactory solutions rather than optimal ones, allowing composition to be performed even on resource-constrained devices.

II. Approach

A. Preliminaries

A *concrete* service cs_i is defined as a tuple [1] $\langle cs_i^{in}, cs_i^{out}, cs_i^{prec}, cs_i^{postc}, cs_i^{QoS}, cs_i^{ctx} \rangle$ that performs a task by acting on input data in to produce output data out, with pre-conditions $prec$, post-conditions $postc$, Quality of Service values QoS, and context information ctx; and an *abstract* service as_i defined as a tuple $\langle as_i^{pre}, as_i^{post}, as_i^{cs} \rangle$ and realized by several concrete services that offer the same functionality ($as_i^{cs} \in \{cs_{(i,1)}, ..., cs_{(i,n)}\}$) such that $\forall cs_{(i,j)}, cs_{(i,k)} \in as_i^{cs} / (as_i^{pre} = cs_{(i,j)}^{pre} \cap cs_{(i,k)}^{pre}) \wedge (as_i^{post} = cs_{(i,j)}^{post} \cap cs_{(i,k)}^{post})$.

B. System Architecture

The *COPERNIC* Agent is a cognitive module inspired by architectural principles defined by the Common Model of Cognition (CMC) [5], [8], a computational model that captures a consensus about the structures and processes that are similar to those found in human cognition. Cognitive modules are (see detailed description in [9]): ***1. Perception:*** it perceives agent's current state by processing both external (e.g., user requests) and internal (i.e., user's context, sensor readings, and QoS) sensory inputs. It outputs a set of percepts (symbolic representation units) such $P = \{p_1...p_n\} \mid \forall p_i \in PR$, where PR is a set of premises such that $as^{pre} \cup cs^{pre} \subseteq PR$. ***2. Working Memory (WM):*** WM holds previous percepts not yet decayed away, and local associations from declarative memories that are combined with the percepts to understand the current state of the service composition. WM defines a limited storage capacity and a recency-based decay function that keeps active a limited number of units, expressed as a base-level activation function. Contents of WM are used as inputs for service matching, i.e., $w_i \subseteq (as^{pre} \cup cs^{pre}) \subseteq PR$. ***3. Declarative Memories:*** the WM cues the declarative memories (i.e., *Episodic Memory (EM)* that retrieves information about services' historic performance, context, etc., and *Semantic Memory (SM)* that retrieves service descriptions, user preferences, etc.) and stores local associations. EM is a content-addressable associative memory represented through a Sparse Distributed Memory; and SM is implemented using a Slipnet, an activation passing semantic network. This module outputs a set of premises $D = \{d_1...d_n\} \mid \forall d_i \in PR$. ***4. Selective Attention (SA):*** SA filters out a continuous stream of content from WM so the agent only focuses on the most relevant information needed for matching abstract services. Goals are decomposed and abstract services compete and cooperate among them in order to get the focus of attention. SA uses a Behavior Network (BN) [6], [7], a hybrid system that integrates both spreading activation dynamics and a symbolic, structured representation. Each behavior of the BN maps to a single abstract service as, and *"service discovery"* emerges from the activation/inhibition dynamics among all services. Abstract services (behaviors) distinguish between expected/non-expected (positive/negative) postconditions (in terms of an "add" and a "delete" list) and define a level of activation as_i^{α}. Also, the model defines 5 parameters to tune the global behavior of the network: π is the mean level of acti-

978-1-7281-3851-0/19/$31.00 ©2019 IEEE
DOI 10.1109/SERVICES.2019.00118

vation, θ is the threshold for becoming active, ϕ is the amount of activation energy a WM unit injects into the network, γ is the amount of energy a goal injects into the network, and δ is the amount of activation energy a protected goal takes away from the network. ***5. Procedural Memory (PM):*** PM defines a set of heuristics to: 1) discover concrete services based on QoS attributes; and 2) adjust the BN parameters to make the global behavior be more adaptive. PM applies the following heuristics [7] to keep the balance between: (1) goal-orientedness vs. situation-orientedness, $\gamma > \phi$; (2) deliberation vs. reactivity, $\phi > \gamma \wedge \phi > \theta$; and (3) bias towards ongoing plan vs. adaptivity, $\phi > \pi > \gamma$. The values are dynamically adapted using a utility-based learning mechanism. ***6. Action Selection (AS):*** AS processes both internal actions (such as goal setting) and and external actions (such as triggering a device's effector/actuator, and invoking the discovery mechanism to execute concrete services). ***7. Cognitive Cycle:*** unlike traditional approaches that create upfront composition plans which are prone to inadaptability, in our approach, plans emerge from the interaction of cascading sequences of **cognitive cycles** corresponding to perception-action loops (modules 1-6) where compositional conditions are validated and reasoning and planning take place. This contribution allows service composition to be more reactive, robust, and adaptive to dynamic changes while composition plans are generated on-the-fly by using minimal resources as a result of filtering out a continuous stream of information.

III. Evaluation

We used the NS-3 simulator to compare the performance of *COPERNIC*[1] against two state-of-the-art decentralized service composition models: GoCoMo, a goal-driven service model based on a decentralized heuristic backward-chaining planning algorithm [2]; and CoopC, a decentralized goal-driven cooperative composition model that does not support runtime composite service adaptation [3]. We modified service density (sparse (SD-S): 20, medium (SD-M): 40, dense (SD-D): 60); composition length (5 services (CL-5) or 10 services (CL-10)); and node mobility (slow (M-S): 0-2m/s, medium (M-M): 2-8m/s, and fast (M-F): 8-13m/s); and we measured 3 different metrics: composition time (**CT** in seconds), average memory used during the composition (**MU** in Kb), and a planning failure rate **PFR** (# of failed planning processes / # of all the issued requests). Table I shows the results (blue/red cells are the best/worst measurements for each category, respectively). In particular, GoCoMo's failure rate was lower than *COPERNIC* (12-38%) when the mobility was slow. This difference dropped to 7-13% in fast-mobility high-density scenarios because *COPERNIC* is less sensitive to mobility changes thanks to service information is stored in the WM and gradually fades away, which means that it can still be accessible even when a service disappears and reappears later in time, allowing the service to promptly participate again in the composition without producing significant planning failures. In comparison with CoopC, *COPERNIC* got 12-25% less failures due to CoopC does not support runtime adaptation and poorly handles mobility changes. Regarding composition time, *COPERNIC* tailored composite services up to 42% and 71% faster than GoCoMo and CoopC, respectively; and it used up to 72% and 84% less memory than GoCoMo and CoopC, respectively. The reason for this significant reduction is that *COPERNIC* is continuously filtering out the stream of incoming information, which keeps it into reasonable margins of resources usage, despite of the dynamism of the environment. It is worth noting that *COPERNIC* did not show a significant difference in memory usage when using a composition length of either 5 or 10 services (-4% - 11%) in comparison with GoCoMo (60% - 190%) and CoopC (157% - 201%), which suggests that our approach could be smoothly scaled up.

[1] https://github.com/ojrlopez27/copernic

TABLE I
FLEXIBILITY OF SERVICE COMPOSITION

			M-S			M-M			M-F		
			SD-S	SD-M	SD-D	SD-S	SD-M	SD-D	SD-S	SD-M	SD-D
COPERNIC	CL-5	PFR (%)	18.2	3.7	1.1	17.5	1.4	1.4	21.1	3.3	1.1
		CT (sec)	0.9	0.5	0.8	1.1	1.2	1.2	1.1	1.4	1.4
		MU (Kb)	63	81	93	67	86	93	73	88	98
	CL-10	PFR (%)	17.8	3.7	1.1	17.7	1.5	0.5	19.7	3.8	1.4
		CT (sec)	1.2	0.6	0.8	1.2	1.2	1.1	1.2	1.3	1.9
		MU (Kb)	70	86	92	70	73	85	78	89	94
GoCoMo	CL-5	PFR (%)	13.1	3.3	0.6	16.1	1.2	0.3	18.0	3.1	0.9
		CT (sec)	1.3	0.7	0.9	1.3	1.4	1.4	1.3	1.3	1.4
		MU (Kb)	79	93	112	78	93	110	80	94	114
	CL-10	PFR (%)	16.2	2.3	0.8	24.7	1.1	0.4	22.1	3.5	1.3
		CT (sec)	2.1	2.2	2.2	2.2	2.3	2.3	2.3	2.4	2.4
		MU (Kb)	213	273	314	201	287	308	221	286	345
CoopC	CL-5	PFR (%)	16.2	2.4	0.8	21.9	1.3	2.3	24.5	3.7	1.2
		CT (sec)	1.8	1.9	1.9	1.9	1.8	2.1	1.9	2.1	2.2
		MU (Kb)	114	245	367	121	239	353	117	275	359
	CL-10	PFR (%)	24.0	2.3	1.3	25.2	2.4	1.2	31.8	4.2	1.6
		CT (sec)	4.1	4.2	4.2	4.5	4.7	4.9	5.0	5.1	5.5
		MU (Kb)	325	476	593	332	488	605	345	497	657

IV. Conclusions and Future Work

We propose a cognitive model that efficiently and dynamically orchestrates distributed services under highly changing conditions. Our approach focuses on bounded rationality rather than optimality, allowing the system to compensate for limited resources by filtering out a continuous stream of incoming information. We tested our model against state-of-the-art service composition models while modifying mobility, service density and composition complexity features, and the results were promising demonstrating that our approach may be suitable for MPC environments where resources are scarce.

References

[1] S. Balzer, "Bridging the gap between abstract and concrete services a semantic approach for grounding owl-s," in *Semantic WS*, 2004.
[2] N. Chen and S. Clarke, "Goal-driven service composition in mobile and pervasive computing," *Services Computing*, vol. 11, no. 1, 2018.
[3] A. Furno, "Efficient cooperative discovery of service compositions in unstructured p2p networks," in *Parallel Processing*, 2013.
[4] A. Immonen and D. Pakkala, "A survey of methods and approaches for reliable dynamic service compositions," *SOA*, vol. 8, 2014.
[5] J. Laird, "A Standard Model of the Mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics," *AI*, no. 4, 2017.
[6] P. Maes, "How to do the right thing," *Connection Science*, 1989.
[7] O. J. Romero, "An evolutionary behavioral model for decision making," *Adaptive Behavior*, vol. 19, no. 6, pp. 451–475, 2011.
[8] ——, "CogArch-ADL: Toward a Formal Description of a Reference Architecture for the Common Model of Cognition," *PCS*, 2018.
[9] ——, "Dynamic Service Composition Orchestrated by Cognitive Agents in Mobile & Pervasive Computing," in *AIMS*, 2019, p. In press.
[10] O. J. Romero and S. Akoju, "An efficient mobile-based middleware architecture for building robust, high-performance apps," in *ICSA*, 2018.
[11] O. J. Romero and A. Dangi, "NLSC: Unrestricted Natural Language-based Service Composition through Sentence Embeddings," in *SCC*, 19.