in different compositions, they rely on the information that W-context caters on the progress of these compositions before these Web services decide on participating in a new composition. It is noted that contexts of Web services have a fine-grained content, whereas contexts of composite services have a coarse-grained content. For illustration purposes, we suggest some arguments that could populate the C-context of a composite service:

- Previous/Current/Next Web services: Indicates which component Web services of the composite service have been executed/are now under execution/will be executed.
- Starting time: Informs when the execution of the composite service has started.
- Status per Web service: Corresponds to the status of each component Web service of the composite service that is deployed (based on status argument of W-context).

Since it is unlikely that a certain provider would deliver all types of Web services and their respective contextual information [4], contexts will be heterogeneous in terms of content, granularity, and structure. To manage this heterogeneity, contexts of Web services must be conciliated at the composite-service level (Figure 3). For example, if the W-context of a Web service that is an element of a composite service has "location of execution" argument and the W-context of a second Web service of the same composite service has "site of execution" argument, these arguments are treated as the same by the composite service. To ensure the differences between contexts' arguments are recognized, context ontology is used. "Location of execution" and "site of execution" arguments here mean the computing resource on which the performance of a Web service happens. The specification of the context ontology calls for a dedicated language such as OWL-C, or Ontology Web Language-based Context. More details on the OWL-C language are given in [5].

It is known in the research community that a Web service can be described along the following three categories (www.daml.org/services/owl-s/): profile, process model, and grounding. By putting forward OWL-C as a specification language for context, it would be appropriate to ensure that the description of context occurs along these three categories. The profile would describe the arguments and capabilities of context (What does the context require and provide?). The process model would suggest how context collects raw data from sensors and detects changes that need to be submitted to the Web service. Finally, the grounding would define the bindings (protocol, input/output messages, among others) that make context accessible to a Web service.

Here, we argued the rationale of associating context with Web services. Context has been used in conjunction with three intertwined steps. The first involved deploying Web services that assess the environment before they accept participating in compositions. The second step involved reducing the semantic heterogeneity gap between independent Web services that have all agreed to participate in a composition. Finally, the third step addressed conciliating contexts of Web services using ontologies. ■

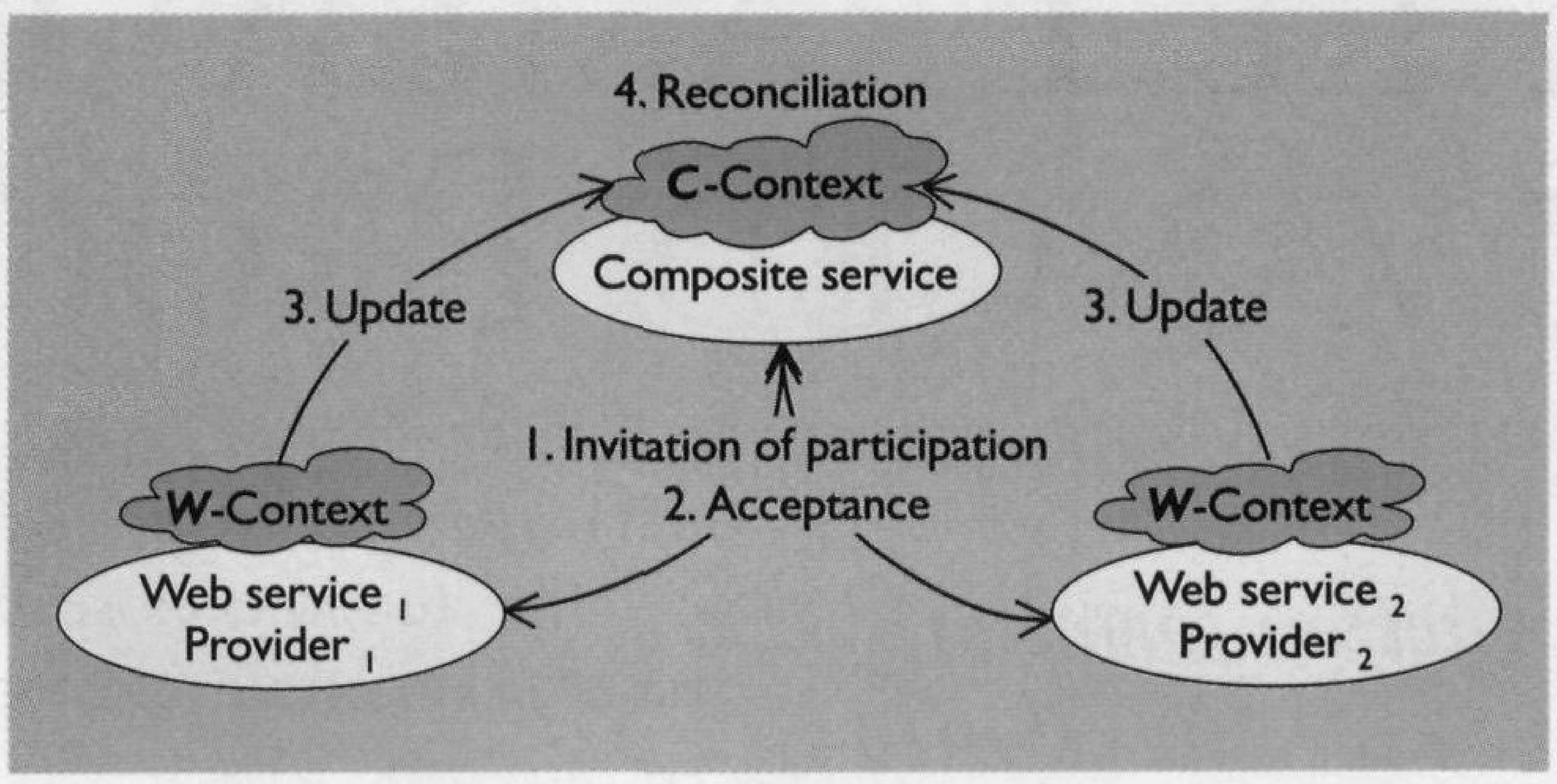

Figure 3. Context and ontology use in Web services composition.

References

1. Benatallah, B., Sheng, Q.Z., and Dumas, M. The self-Serv environment for Web services composition. *IEEE Internet Computing 7,* 1 (Jan./Feb. 2003).
2. Birman, K.P. Like it or not, Web services are distributed objects. *Commun. ACM 47,* 12 (Dec. 2004).
3. Brézillon, P. Focusing on context in human-centered computing. *IEEE Intelligent Systems 18,* 3 (May/June 2003).
4. Hegering, H.G., Küpper, A., Linnhoff-Popien, C., and Reiser, H. Management challenges of context-aware services in ubiquitous environments. In *Proceedings of the 14th IFIP/IEEE Workshop on Distributed Systems: Operations and Management* (Heidelberg, Germany, 2003).
5. Maamar, Z., Narendra, N.C., and van den Heuvel, W.J. Toward an ontology-based approach for specifying contexts of Web services. In *Proceedings of the Montreal Conference on e-Technologies* (Montreal, Canada, 2005).
6. Maamar, Z., Kouadri Mostéfaoui, S., and Mahmoud, Q.H. On personalizing Web services using context. *International Journal of E-Business Research (Special Issue on E-Services). Idea Group 1,* 13 (July–Sept. 2005).
7. Papazoglou, M. and Georgakopoulos, D. Service-oriented computing: Introduction. *Commun. ACM 46, 10* (Oct. 2003).
8. Satyanarayanan, M. Pervasive computing: Vision and challenges. *IEEE Personal Communications 8,* 4 (Aug. 2001).
9. Sciore, E., Siegel, M., and Rosenthal, A. Using semantic values to facilitate interoperability among heterogeneous information systems. *ACM Trans. Database Systems 19,* 2 (1994).

Zakaria Maamar (zakaria.maamar@zu.ac.ae) is an associate professor at Zayed University, Dubai, U.A.E.
Djamal Benslimane (djamal.benslimane@liris.cnrs.fr) is a professor at Claude Bernard Lyon 1 University, Lyon, France.
Nanjangud C. Narendra (narendra@in.ibm.com) is a research staff member at IBM India Research Lab, Bangalore, India.