Dynamic Variability Meets Robotics

Davide Brugali,
University of Bergamo, Italy

Rafael Capilla
Rey Juan Carlos University, Spain

Mike Hinchey,
Lero - the Irish Software Research Center

Keywords: Software technologies, dynamic variability, robotics

The cost of creating new robotics products significantly relates to the complexity of developing their software control systems, which must be flexible enough to easily accommodate volatile requirements or changing needs. Today, context-aware engineers focus on the inclusion of contextual capabilities as an essential part of dynamic behavior, and the development of embedded software-intensive systems with context-aware and runtime capabilities[1] offers a good mechanism for addressing the dynamic changes and evolution scenarios modern robotic systems demand.

ROBOTS AS CONTEXT-AWARE SYSTEMS

Robots of the 21st century are versatile machines that have the potential to enhance safety in transportation, reduce the use of pesticides in agriculture, and to improve efficacy in the fight against crime and civilian protection. Today, one the barriers in robotics development is the challenge of overcoming the significant cost of developing new robotics products, which is significantly related to the complexity of developing software control systems that are flexible enough to easily accommodate volatile requirements or changing needs.

Robot software development demands advanced cognitive capabilities, such as perception, planning, monitoring, coordination, and control, in order to cope with unexpected situations reliably and safely. Take for example the area of agricultural robotics, where many interacting factors govern the effectiveness of herbicides and the potential for crop injury. Such contextual capabilities often include: (a) environmental conditions, such as temperature, and relative humidity, weed species present in the field; (b) task execution factors, such as depth of planting, time of planting, time of herbicide application; (c) required resources, such as type of herbicide, type of robotic equipment and agricultural equipment.

All of these factors may affect the robotic system’s configuration both at deployment-time and runtime and, ideally, robots should be able to manage their context-aware capabilities dynamically and address both possible and unforeseen variations during execution. Because a large number of parameters characterize the possible variations in robotic configuration tasks, handling this variability in robotics remains challenging for both software engineers and robot operators. In addition, the runtime and adaptation requirements robots need adds an extra level of complexity to managing the diversity of scenarios when context changes.

Stringent robotic runtime requirements and adaptation challenges

Facing the complexity of open-ended environments, robots must guarantee an appropriate level of quality of service (QoS) in complex operational contexts. Typical non-functional properties that an autonomous robot should exhibit are: robustness to mange exceptional situations, performance using limited resources (e.g., tiny memory model), and long-term continuous operation.

In robotics, operational contexts are real-world scenarios consisting of complex tasks robots must perform in an open-ended environment. This requires that the robot has the ability to exploit the available resources in the best way possible. For example, a GPS localization system cannot be used inside a building, while the performance of a stereo vision system is affected by the environmental lighting conditions.

In many cases, runtime adaptation of the robot control system requires some tuning of the control parameters, replacing its functionality, or modifying the coordination policy. Just to give an example, different settings of the robot dynamics might be associated with features representing common behaviors (e.g., high performance, safety) or use different motion planning algorithms in environments with fixed or moving obstacles (e.g., people). From this perspective, we identify the following challenges that modify the robot's runtime behavior:

- Robot reconfiguration tasks using limited resources, as part of the robot functionality is replaced or updated at runtime.
- Continuous adaptation and evolution of the robot behavior where new capabilities can be added at runtime (e.g., update the database to recognize a new object using the robot's current context-aware capabilities).
- Dynamic optimization of robot tasks (e.g., use a shorter new path to arrive at a given point).
- Change the states of context properties dynamically (e.g., allocating a battery charging station to a mobile robot does not automatically enable the mobile robot to start recharging its battery from a distance).

THE ROLE OF DYNAMIC VARIABILITY

In order to address the aforementioned challenges, in the last decade Model-driven Engineering (MDE) approaches have been increasingly introduced into robotic systems. As an example, the HyperFlex toolkit [2] supports the development of robotic software product lines by providing tools for modeling, composition, and resolution of robotic variability.

Other recent approaches such the emerging paradigm of Dynamic Software Product Lines[3] (DSPLs) offer a variety of techniques suitable for coping with modern robot needs. One of the key techniques DSPLs uses to manage the adaptation challenges at runtime is dynamic variability, which offers a way to modify the structural variability and context features at runtime[4].

Recently, runtime variability has attracted the attention of robot designers as it allows for the addition or modification of robot behavior and context capabilities dynamically and addresses unforeseen scenarios more effectively with less human intervention. The capability to offer late binding time of system features offered becomes quite suitable for robotics reconfiguration and redeployment. Thus, providing features on-demand at different granularity levels eases the adaptation of on-board robot capabilities that can be composed and activated dynamically.

Representing context variability

One of the current challenges facing dynamic variability is how to represent and manage the variety of context properties of robots. Recently, the role of context variability, as a technique that combines variability modeling with context properties, is gaining popularity for handling the variations that occur in many context-aware and self-adaptive systems. New design and development approaches like context analysis has become a key activity for modeling context properties in autonomous and self-adaptive systems under the scope of a DSPL approach. Other complementary initiatives supporting context modeling is context-oriented programming (COP) language[5]. COP languages such as JCop provide explicit constructs that enable the execution of context-dependent behavioral variations. From the perspective of context variability modeling, COP languages should offer specific characteristics for: (i) modeling context properties, (ii) supporting reflection and runtime behavior and modification, (iii) distribution and security aspects, (iv) multiple and dynamic binding, and (v) feature interaction.

ROBOTICS SOFTWARE DEMANDS DYNAMIC VARIABILITY SOLUTIONS

Today, dynamic variability techniques offer a good solution for cope with the stringent runtime requirement of robotic systems. However, there are two main challenges for robotics software engineers when they need to manage contextual properties dynamically. The first challenge deals with the representation of the context variability and the second deals with the way by which the structural variability is managed dynamically.

Modeling robot contextual properties

The design of variability of robots is challenging, as many context properties change their state or the robot requests a new functionality or feature at runtime. From previous work[6] we selected a strategy to represent context and non-context properties in one single feature model. This representation schema simplifies the description feature model and reduces the number of dependencies between context and non-context features.

In Figure 1 we use a context feature model to represent the robot's functionality for robust navigation of a mobile manipulator. These features enable *transportation* tasks of typical robocup@work challenges (http://www.robocupatwork.org/). For instance, the *mechanical embodiment* of the robot (i.e., Omnidirectional and Differential Drive) is a typical non-context feature that designers model at deployment time, when the control system of the robot is deployed on a specific robotic platform. In contrast, the *Motion Behaviour* is a context property since it depends on the type of load that the robot has to carry (e,g., moving a liquid demands a smooth motion behavior by the robot). The designer can define different categories or classifiers to group related features that pertain to different context capabilities.

Robot dynamic operations

The second challenge refers to the mechanisms robots need to manage the contextual variability dynamically. Current approaches enable adding, removing or changing single features at runtime on behalf of the classifiers that allow to, for instance, plug a new feature at runtime. Because dynamic adaptation requires a reconfiguration of the control system beyond the simple activation or deactivation of system parameter, robots must support the replacement of a complete subsystem encompassing groups of related features. We address this scenario in Figure 1, where the navigation functionality can switch between two alternative strategies: *a map-based and a marker-based navigation*. The map-based navigation allows the robot to plan an obstacle-free path from the current location to the target location using a detailed geometric representation of the environment. The robot uses a laser scanner to recognize places and locate itself with respect to the map. Alternatively, the marker-based navigation tracks a sequence of visual landmarks placed on the floor and detected by the robot's webcam.

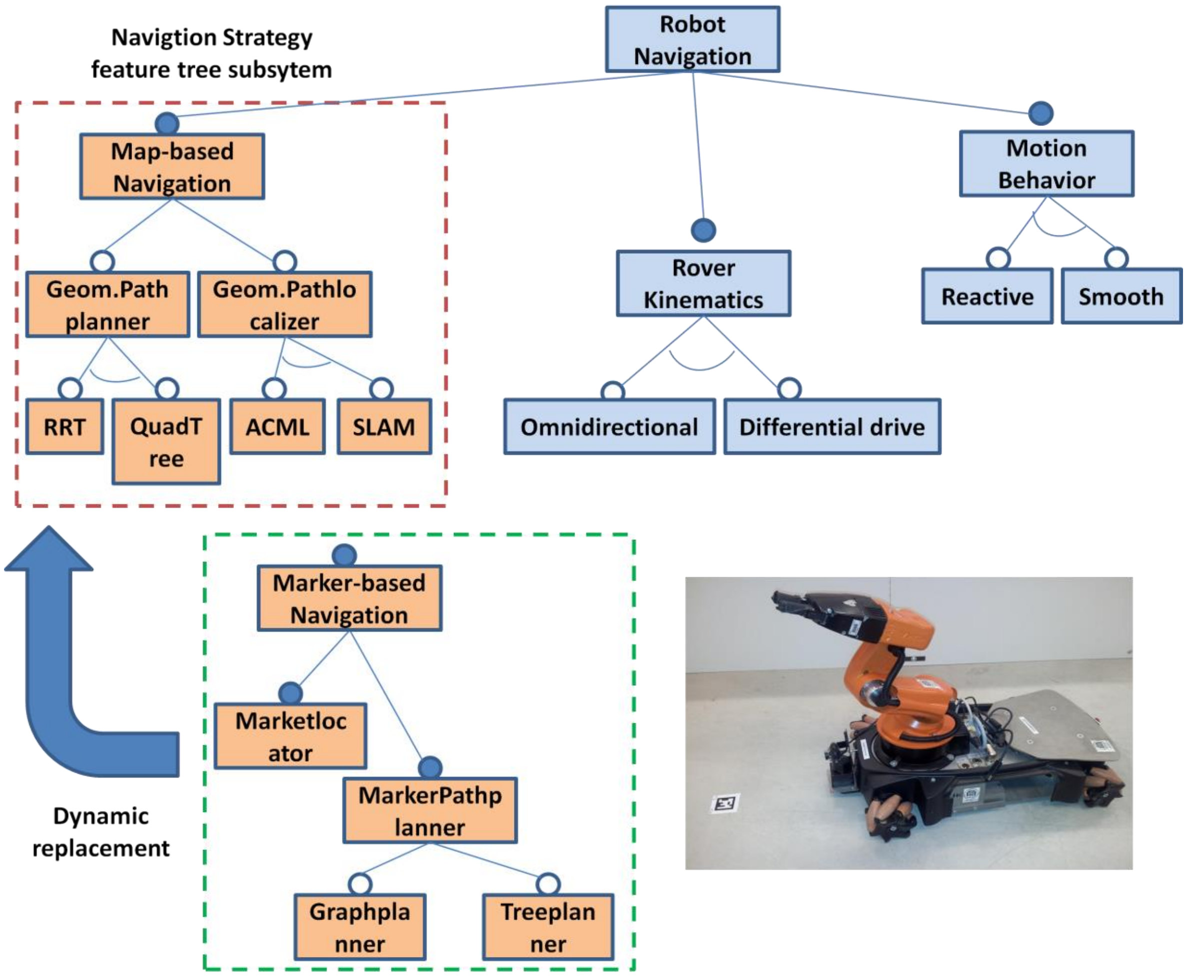

Figure 1: Feature model of a robot mobile manipulator. Modifications to the structural variability at runtime can replace single and groups of feature dynamically (e.g.: navigation functionality).

The former strategy is more flexible but vulnerable to map inaccuracies while the latter is more robust, but it requires structuring the environment artificially. When the robot travels through different regions and a map is not available or the markers are not visible, a replacement of the navigation strategy becomes necessary. In such a case, dynamic variability replaces the current navigation model by another and, hence, reconfigures the system accordingly. The classifiers that group different categories of features make the dynamic replacement of single features and groups of features supporting the new functionality possible.

Another important aspect in relation to the dynamic behavior is the time when variants bind to new values during reconfiguration operations. Allowing multiple binding times in robots (e.g., reconfiguration, testing, runtime as normal operational mode, etc.) for different types of features enable the definition of different transitions between different operational modes in the robot which makes adaptation more easy and "smart".

Opportunities for dynamic behavior

Dynamic variability offers a good choice for addressing the challenge for smarter adaptation of robots in unforeseen scenarios. Predicting and managing the evolution of robot behavior is still hard, and the many situations where robots perform tasks autonomously can be enhanced through the use of dynamic software product lines techniques. Dynamic variability offers new opportunities and capabilities to provide: (i) add/remove/change features dynamically, (ii) transition between different operational modes, (iii) better collaborative opportunities in robotic swarm projects, (iv) manage context features more efficiently, and (v) reduce the burden of remote operations.

CONCLUSION

The robotics area is still a green but promising field for testing dynamic variability approaches. As robots are becoming more and more smart, collaborative, and autonomous, the variety of changing scenarios that use context information to adapt a robot's behavior and support human tasks increases every day. Consequently, dynamic variability plays a significant role in enhancing robot adaptation capabilities.

References

1. J. Bosch and R. Capilla, "Dynamic Variability in Software-Intensive System Families", *Computer* vol. 45, no. 10, 2012, pp. 28-35
2. L. Gherardi and D. Brugali, "Modeling and Reusing Robotic Software Architectures: the Hyperflex Toolchain", *IEEE International Conference on Robotics and Automation*, 2014
3. S. Hallsteinsen, M. Hinchey, S. Park and K. Schmid, "Dynamic Software Product Lines", *Computer*, vo. 41, no. 4, April 2008, pp 93-95
4. R. Capilla, J. Bosch, P. Trinidad, A. Ruiz-Cortés and M. Hinchey, "An Overview of Dynamic Software Product Line architectures and techniques: Observations from research and industry", *Journal of Systems and Software* vol. 91, no. 5, 2014, pp. 3-23
5. P. Constanza and R. Hirschfeld, "Language Constructs for Context-oriented Programming: An Overview of ContextL", *Dynamic Languages Symposium*, 2005, pp. 1-10
6. R. Capilla, O. Ortiz and M. Hinchey, "Context Variability for Context-Aware Systems", *Computer* vol. 47, no. 2, 2014, pp. 85-87

Davide Brugali is Associate Professor of Computer Science at University of Bergamo, Italy. Contact him at brugali@unibg.it.

Rafael Capilla is Associate Professor of Computer Science at Rey Juan Carlos University, Spain. Contact him at rafael.capilla@urjc.es

Mike Hinchey is Director of Lero–the Irish Software Research Center and Professor of Software Engineering at University of Limerick. Contact him at mike.hinchey@lero.ie