Continuous Adaptation Management Viewpoint for Collective Intelligence Systems

CIS-ADAPT Architecture Viewpoint Description

From the data collection and analysis and our own experiences with CIS design and development, we developed a novel architecture viewpoint aligned with the ISO/IEC/IEEE 42010 standard which provides an adaptation-specific view on CIS architectures and is implementation agnostic. The *architecture viewpoint for continuous adaptation management in collective intelligence systems (CIS-ADAPT)* frames the essential concerns of stakeholders with an interest in handling CIS-specific adaptation across the system's life cycle, starting from its inception and during its operation. The viewpoint defines a set of four model kinds for identifying, designing and realizing adaptation in CIS key elements and aims to support software architects in the system design. It unifies CIS-specific aspects with established adaptation approaches so that this approach represents a useful addition to domain-specific adaptation approaches. It is important to note that the focus of the viewpoint is on CIS-specific adaptation and its impact on the system architecture. As such, architects may use additional architectural approaches, such as additional viewpoints or patterns, to deal with adaptation in traditional software system elements and other stakeholder concerns.

The architecture viewpoint is structured using the standard template of the ISO/IEC/IEEE 42010 standard:

- Name: set of words to refer to the viewpoint
- Overview: short description of the viewpoint
- Stakeholders: individual, team, organization with an interest in the concerns addressed by the viewpoint
- Concerns: interests in the CIS relevant to stakeholder(s) which are expressed in form of questions to help stakeholders framing their concerns
- Model kinds: conventions for a type of model
- Metamodels: core constructs of the model kinds
- Analysis: methods to check, reason about, transform, predict, apply and evaluate architectural results from the view generated from this viewpoint

In the following section, we provide a listing of identified stakeholders of the architecture viewpoint and highlight their roles in a CIS environment. We also present the identified adaptation concerns that are framed by the viewpoint. Then we describe each kind of model and its metamodel used in this viewpoint as well as analysis methods in detail.

A. Stakeholders and Concerns

Table I shows an overview of the identified stakeholders and their concerns. Stakeholders of the architecture viewpoint include:

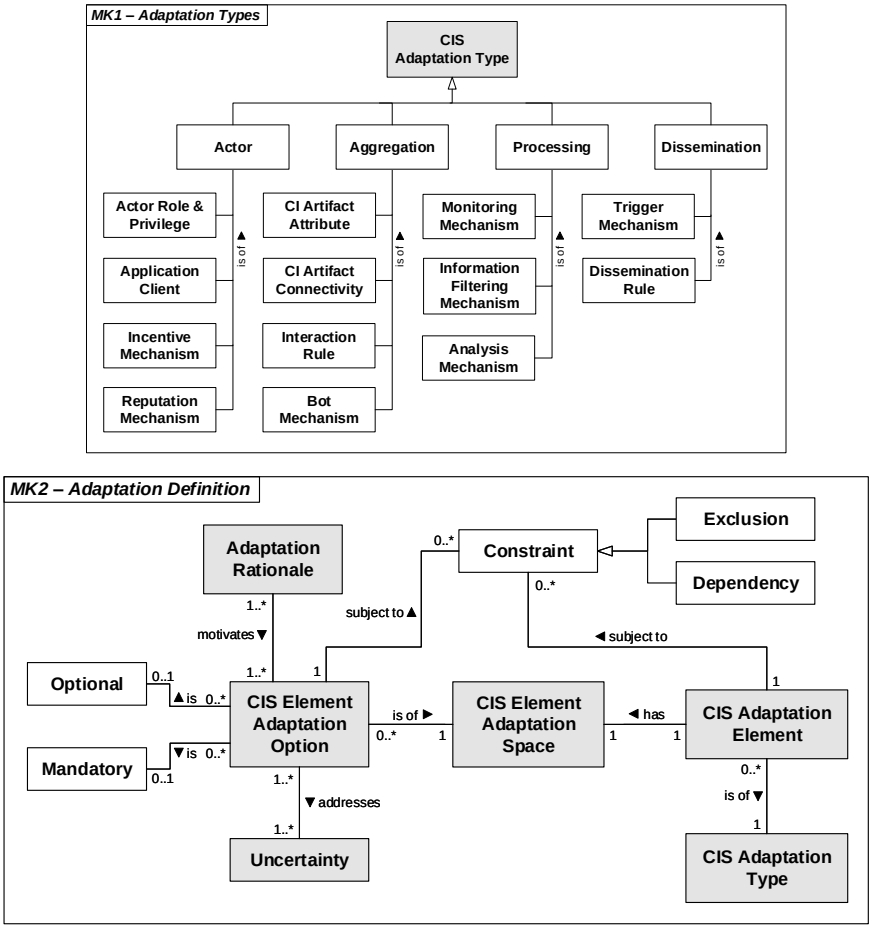
- *architect(s)* who design and describe the CIS architecture and identify the commonalities and the adaptation space in the system,
- *owner(s)* who define the CIS's purpose and business goals and operate it to provide the service to the users,
- *manager(s)* who are responsible for overseeing CIS operation, and
- *analyst(s)* who monitor the CIS, perform analysis of its behavior and assess the performance of the system in terms of quality criteria.

This viewpoint particularly focuses on the technicalities of adaptation management in CIS, which are no direct concerns

TABLE I. Continuous Adaptation Management Viewpoint for CIS - Overview

Name	Continuous Adaptation Management Viewpoint for CIS
Overview	The architecture viewpoint deals with the main stakeholder concerns related to the continuous management of CIS-specific adaptation and defines models for the identification, design and realization of adaptation elements and their space of possible options across the system's life-cycle. The models show the relevant architectural information that is essential to guide a successful preparation for anticipated changes in the system's environment or requirements.
Stakeholders	*Architect(s)* who design and describe the CIS architecture and identify the commonalities and the adaptation space in the system. *Owner(s)* who define the CIS's purpose and business goals and operate it to provide the service to the users. *Manager(s)* who are responsible for overseeing CIS operation. *Analyst(s)* who assess the performance of a CIS in terms of quality criteria.
Concerns	*C1 - Adaptation Identification:* How can adaptation be exploited to enhance the operation of a CIS? What are possible adaptation elements in a CIS? What are the implications of adaptation elements in the design of a CIS? *C2 - Adaptation Management:* What options are available to resolve an adaptation element? What are the effects of different options? What are dependencies between different adaptation elements and options? When are adaptation elements resolved? Who is responsible for handling the adaptation and selecting adaptation options? *C3 - Adaptation Evolution:* When are adaptation activities be performed in the CIS life-cycle? How does adaptation influence the CIS evolution?

TABLE II. CONTINUOUS ADAPTATION MANAGEMENT VIEWPOINT FOR CIS - MODEL KINDS

Model Kinds	*MK1 - Adaptation Types (deals with concern C1):* A model that describes *where* adaptation can likely be achieved in a CIS to address uncertainties by identifying potential points of adaptation in CIS-specific system areas along with possible alternatives. *MK2 - Adaptation Definition (deals with concern C2):* A model that clarifies *what* adaptation is about in the CIS-of-interest and describes details about the identified adaptation elements selected for adaptation, the associated element adaptation space of options to address particular uncertainties, and what constraints are applied on the relations between adaptation elements, options, and adaptation elements and options. *MK3 - Adaptation in Time (deals with concern C3):* A model that describes *when* adaptation activities are applied by responsible entities and how adaptation evolves across the CIS's life-cycle. *MK4 - Adaptation Workflow (deals with concern C2):* A model that describes *how* the adaptation elements are realized and resolved, and who is responsible for selecting adaptation options and triggering the changes.
Metamodels	

Key: UML

of system users, who contribute continuously to it. Thus the users are no stakeholders in terms of this viewpoint, but they are certainly affected by the design decisions made by applying this viewpoint.

The viewpoint addresses the following CIS-specific adaptation concerns of the stakeholders: *adaptation identification*, *adaptation management*, and *adaptation evolution*.

B. Viewpoint Model Kinds

The viewpoint comprises four model kinds presented in Tables II and III: *adaptation types*, *adaptation definition*, *adaptation in time* and *adaptation workflow*.

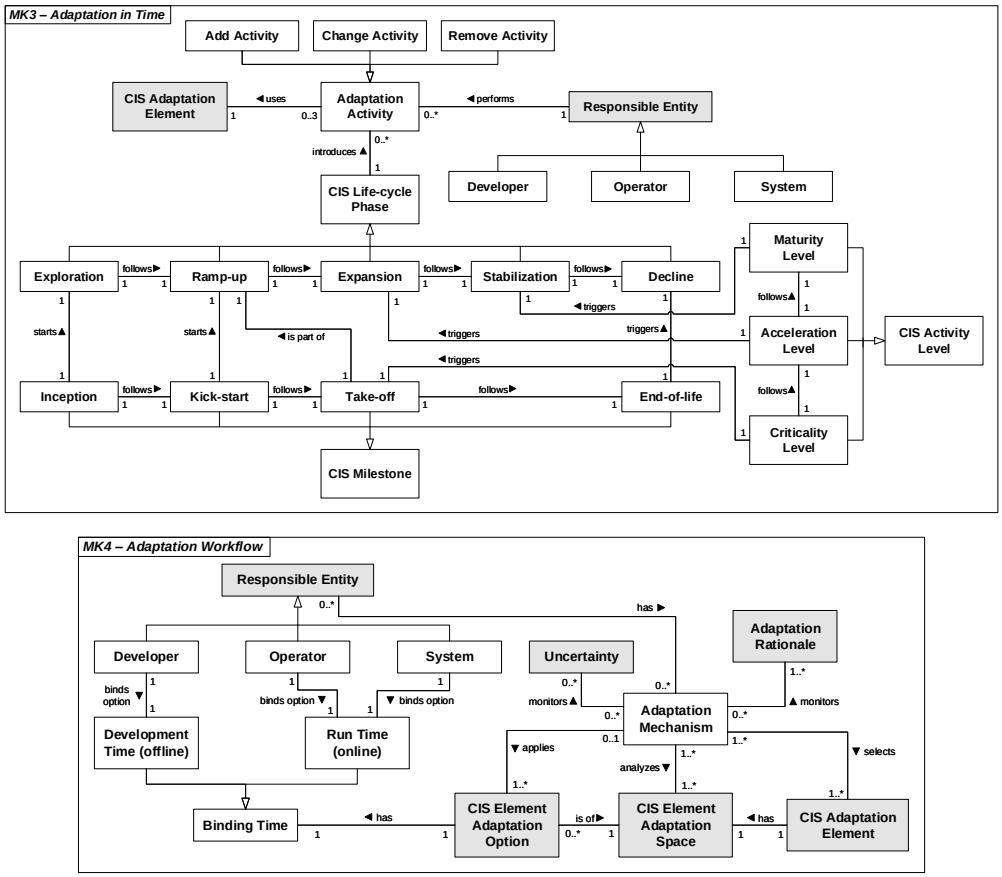
1) Adaptation Types Model Kind: This model kind describes the subject of adaptation, comprising four CIS-specific adaptation types along with adaptation elements: (1) *Actor*, (2) *Aggregation*, (3) *Processing*, and (4) *Dissemination*. Examples of concrete options of the adaptation element *Incentive Mechanism* of the type *Actor* can be: awarding badges, up-votes, and likes. Concrete options for the adaptation element *Dissemination Rule* of type *Dissemination* are artifact change reports, weekly digests, and monthly personal recommendations. This model kind supports software architects with defining what adaptation types and related adaptation elements are relevant to implement in the context of the specific CIS-of-interest based on the concretely identified adaptation types.[1]

2) Adaptation Definition Model Kind: This model kind describes *what* adaptation is. It defines the possible adaptation options of an adaptation element, i.e., the adaptation space,

[1]Gray shaded boxes in model kinds represent links between multiple model kinds.

TABLE III. CONTINUOUS ADAPTATION MANAGEMENT VIEWPOINT FOR CIS - MODEL KINDS (CNTD.)

Metamodels



Analyses

A1 - Adaptation Effect Analysis (using MK1 and MK2): Assesses the effects of different adaptation option selections on the activities of the system and the actor base using a set of scenarios.

A2 - Adaptation Option Conflict Analysis (using MK2, MK3 and MK4): Reviews the relations and dependencies between adaptation elements and their spaces of options that are simultaneously deployed and bound in different life-cycle stages.

each option representing a particular setting of the element. An adaptation element and its adaptation options are subject to constraints, i.e., they can exclude one another or may have dependencies, e.g., only actors with editor role can activate an artifact protection mechanism. A CIS element adaptation option can be optional or mandatory. Adaptation is then defined as addressing uncertainties by selecting adaptation options for elements according to the adaptation rationales (goals). For instance, a lack of actor attention for specific artifacts observed during operation (uncertainty) may be handled by activating an awareness trigger (adaptation option) to increase contributions to these artifacts (rationale).

3) Adaptation in Time Model Kind: This model describes *when* adaptation can be applied throughout a CISs life-cycle in five phases: (1) *Exploration* phase, (2) *Ramp-up* phase, (3) *Expansion* phase, (4) *Stabilization* phase, and (5) *Decline* phase. Any responsible entity can perform adaptation activities, i.e., add, change, or remove activities to an adaptation element (by adapting its adaptation options) in different phases of the CISs life-cycle. For instance, the operator introduces a monitoring mechanism aiming to identify irregular activities in expansion phase. This activity can be affected by reaching a certain CIS milestone (e.g., after take-off milestone has been reached) or activity level (e.g., when criticality level has been reached). If an adaptation element is not relevant anymore, a responsible entity can remove it, e.g., the system may turn on a dissemination rule when user activity is decreased with some degree over a period of time.

4) Adaptation Workflow Model Kind: This model kind describes *how* CIS-specific adaptations are realized. The adaptation workflow is realized by an adaptation mechanism associated with a responsible entity. A responsible entity can be a developer, an operator, or the system. A developer can apply adaptations offline (that may then be deployed on the running system), while an operator and the system can apply adaptations online. An adaptation mechanism realizes a feedback loop. Concretely, the mechanism monitors uncertainties and checks whether the system complies with its goals (rationales). If the system goals may be jeopardized, the adaptation mechanism performs an analysis of the adaptation space (i.e., the options available for adaptation) of the adaptation elements that can be used to mitigate the uncertainties. Based on this analysis, the adaptation mechanism selects adaptation options for adaptation elements. These adaptation options are then

applied to the system.

C. View Analysis

1) Adaptation Effect Analysis: This analysis uses a set of scenarios to assess the effects of selecting different adaptation options on the behavior of the system and the actor base. The analysis results helps identifying improvements of the adaptation elements and their adaptation options. The results can also provide insights in the conditions when selected options may improve or degrade the CIS behavior, e.g., in the form of increase/decrease of user activity. In the exploration and ramp-up phases, adaptation effect analysis can be done using simulation or via tool-assisted user testing. In later-stage phases further approaches like A/B testing and/or feature toggles can be added to enable automated, data-driven processes for performance analysis, simulation and selection of adaptation options.

2) Adaptation Option Conflict Analysis: This analysis performs a review of the relations and dependencies between adaptation elements, options, and adaptation elements and options that are simultaneously deployed and bound in the different stages of the CISs life-cycle. The results of this analysis help to identify possible conflicts and inconsistencies between CIS adaptation elements/options that need to be resolved. In early stage phases, conflict detection and resolution can be performed manually by the architect by using the CIS-specific adaptation definition and work- flow models. In later stages tool-supported advanced automated approaches such as feature-to-code traceability and consistency checking of the CIS adaptation models are necessary to make conflict identification and resolution viable.