ABSTRACT:

The techniques of programming and methodologies strongly evolved throughout the history of data processing with the evolution of the software systems, these systems indeed tend to become increasingly complex. Component-Based Software Development proved its interests in the control of the complexity of the conceived software, and became a critical factor in the success of development of the software projects by facilitating the maintenance and the evolution of the software and authorizing the development of the bulky systems in terms of size but also of complexity. This style of programming promises the re-use, but is confronted with the problems of code scattering and tangling. The application of Aspect-Oriented Programming on the software components makes it possible to face these problems. Programming called by aspect allowing managing, in a modular way, these concerns by separating them from the basic code. Aspect-Oriented Programming, a new paradigm of the programming which made possible to simplify the writing of the programs data-processing, while making them more modular and easier has to make evolve. Today, the software Aspects and components are two very promising paradigms; who support the re-use and simplify the software development. To date, implementation the simultaneous of these two paradigms remains a field of research very slightly explored. To date no model of component supports in an explicit way the aspects and several questions remain open. Among them: How to integrate the representation of the aspects in the software components? How to manage the interactions and overlappings between aspects? We present in this paper 3ADL, an extension of the model of component IASA defined in the laboratory LRDSI which supports the Aspect-Oriented Programming. This extension consists in equipping approach IASA with the aspect components and aspect ports. The objective of work is to make supports to the model of component IASA the concept of aspect in its entire dimension: Once this concept supported, an architect could define his own Aspect components which it instantiated in the part controls of a component.