

On The Construction of Reflective System Architectures

Jan de Meer

GMD-FOKUS, Kaiserin-Augusta-Allee 31, D-10589 Berlin, Germany; e-mail: jdm@fokus.gmd.de

Abstract: According to [Blair98] a "*reflective System*" is one, that is able "*to inspect and to adapt its own internal behaviour*". This, however is in accordance with controlled, or more precisely with feedback systems. Feedback systems are superpositioned systems consisting of a controlled and of a controlling part. Whereas the controlling part inspects continuously the controlled part and generates advises of how the controlled part must be adapted in order to achieve a given targeted behaviour. Feedback and thus reflection are to be understood as recursive cause-and-effect behaviour. This raises questions of system stability and system sensitivity that must be answered for the construction of reflective system architectures.

The self-reflection capability of feedback systems do also exist in autonomous systems. Whereas the autonomous systems require reflective capabilities to perform autonomously a certain task in e.g. a poisoned environment which is not observable by humans, the reflective systems need the adaptation capabilities to provide with e.g. a guaranteed qualities of services or to eliminate disturbances during operation.

Keywords: Continuous Behaviour, Monitoring, Adaptation, End-to-End Flow Control, System Theory, Autonomous Systems

1. The Constituent Elements

Multimedia applications that operate on the "*information highway*" require the permanent real-time control of their performance. Multimedia flows are considered as processes which continuously generate and consume masses of discrete units of various media, i.e. flows. Therefore, continuous flow processing functions must be build into any end-to-end application, for which the control of QoS characteristics becomes indispensable. The controlling of QoS depends however, from the varying loads and disturbances generated by the "*information highway*" on which the applications run. Hence the environment of the application must be inspected and if necessary either the application or even the information highway must be adapted such that the influences issued by loads or disturbances can be smoothed. Smoothing requires tools for the execution of continuous activities, like inspection, feedback of information or reconfiguration capabilities for adaptations.

A new anticipated architectural model that integrates continuity, feedback and adaptability, the three basic but separate concerns for the construction of reflexive systems, comprises as architectural building blocks "*continuous functions*" [Broy 93a,b, 94], "*open or closed loops*" [Kuo94], [Sch93], and the concept of coupled components of the object-oriented approach from [IS10746] respectively.

A continuous function defines semantically the behaviour of coupled components by continuously separating discrete elements from input flows and by continuously generating discrete elements added to output flows. Continuity among coupled components is achieved as long as a set of predicates defined on a set of continuous variables do not violate some contracted (see ODP enterprise viewpoint in [IS10746]) QoS constraints. The QoS state vector, comprising the continuous variables, describes the current state of consumption of the resources that are assigned to the component.

Both, open or closed loops, are reflective notions. But, by which a clear distinction is made between the controlling and the controlled task. Whereas open loops do not allow to control system errors or loads, closed loops do by inventing a feedback component. Feedback link and the controlling component can both be composed into a com-

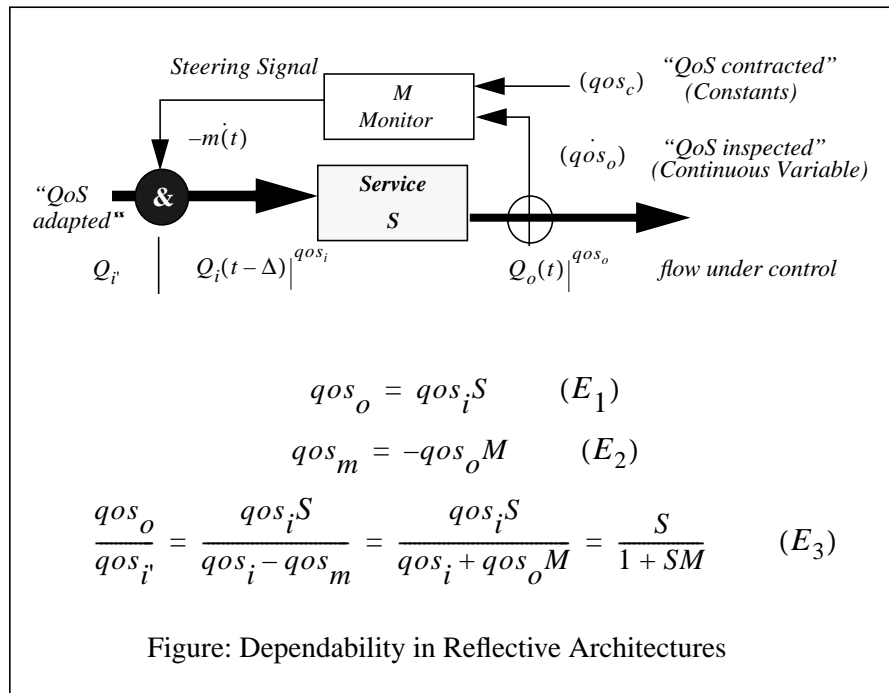
mon monitor component. A monitor - embedded in a closed loop - allows to keep the QoS of the controlled component stable according to a given target. For example, it can keep the presentation of a video flow constant, independent of the various bandwidths available along the transmission path.

2. Fitting things together

For the construction of a reflective architecture, i.e. architectures that include feedback, behavioural aspects of components - i.e. continuity, capacity, performance gradation - must separately be handled. For each aspect a continuous equation will be derived from an ideal loss-less, non-distorted system.

By the chosen methodology, firstly, continuity expresses the fact that any flow generated by some source will occur at its designated sink after a certain delay but unchanged in format and quantity. This observation allows us to calculate the balance of various flows in a system. Secondly, capacity expresses the fact, that any flow inserted into the system will result into the sum of the flow units sent after time t . This observation allows us to calculate the amount of flow units accumulated by the system after time t . Thirdly, performance gradation expresses the fact, that along the path of a flow the quality of the flow will eventually degrade. This is, because flow processing components have to pay a price for their operating. In other words, in direction of the flow the quality of the flow cannot become better provided there is no adaptation. The latter observation allows us to calculate the gradation of changes in quality along the flow.

Controlling is a system-wide activity that comprises all components involved in processing or transmission tasks of the application considered. The task of controlling can be achieved if appropriate auxiliary inspection and activation tools can be attached to the interfaces of a chain of components serving a flow. These tools feedback observations taken from visible interfaces and generate steering signals by which (re-)source of the application are activated or deactivated. Feedback Loops cause interferences among the controlled and controlling components. The behaviour of controlled and controlling components influence each other. This effect is called dependability and can be calculated by relating causing and effecting QoS variables of a system or component. These principles of reflection for the construction of the new middleware generation are outlined in the figure below.



In the figure, it can be seen, that the service providing component S of a reflective architecture are clearly separated from the service controlling component, called monitor M . Whereas the service providing component transports (multimedia) information from left to right, the service controlling component transports (controlling) information from right to left. One can say that the two systems are superpositioned and thus have to cooperate such that the overall system target, possibly captured by a QoS contract, will be achieved under all circumstances arising from loads or distortions. By adopting this view of reflection, components are considered as quality transformers. The QoS provided at the output of a component can be compared to the QoS accepted at the input of the same component. Examples that illustrate the output-input ratio are: A channel obeys a transition delay. The ratio of QoS of the arrival time of flow units at the channel's output to the QoS at the channel's input varies over time; or, if the operation of a channel is distorted the full bandwidth cannot be used by the customer; or, a display device has distorted pixels. So, not every flow unit that will be received by the device will also be displayed.

More formally, the composite reflective architecture provides a service S . This service shall be inspected by the reflective monitor M . It is assumed that all components of this system operate continuously without malfunctions but may change QoS. Hence, the QoS at a component's input can differ from the QoS at its output because there is an internal overhead caused from the processing of the flow. The flow which is to be controlled is designated by Q . The serving component S needs the time Δ to process the flow. Hence, at time $t-\Delta$ the same element of the flow that occurs with quality qos_i at the input is observed at time t at the output but with quality qos_o .

System theory provides with means to amplify or to diminish the QoS of a component, which depends from the component's internal capabilities, e.g. of S or M . Equation E_1 of the figure shows the ratio of output-input QoS as it is generated by the serving component S . Similar, equation E_2 shows the output-input QoS ratio of the feedback component. Because of the anticipated smoothing, which is an inverse effect, the feedback signal obeys a negative sign which means: A raising value of a certain QoS parameter which is not wished, will result in a diminishing steering signal, a decreasing value of the QoS parameter which is not wished as well, will result in an enhancing steering signal to the input QoS of the serving component S . The continuous steering signal is simply added point by point to the QoS at the input of component S .

For example, assume the input- output bandwidth ratio of some component is to be controlled. So, e.g. a "diminishing-bandwidth-coded" steering signal urges the input to the serving component to proceed with a lower bandwidth. So, as long as the monitor produces steering signals the bandwidth will continuously be decreased by fixed steps. In a sense of fuzzy logic, the tasks of inspection and activation may use coded pulse flows in order to adapt a deviated application flow to some given targets.

The variables $m(t), qos_o(t)$ are called continuous variables because they keep values that change continuously. The first variable is the continuous steering signal by which the behaviour of the input flow of S will continuously be influenced. The second variable is the continuous measure of a certain QoS characteristics, e.g. of the bandwidth provided. The continuous steering signal can be understood as a process that continuously issues discrete signals as long as a given target value of the variable has not been achieved.

The QoS resulting from the super positioning of both the serving and the monitoring component is gained by the ratio of the output of component S and by the steering difference between the input qos_i and the steering signal qos_m presented by equation E_1 . If the total QoS of the superpositioned system SM is -1 the system becomes unstable, because the denominator of the quotient of E_2 results to zero. However for $SM \geq 1$ the total system remains stable.

These simple considerations show, that reflective systems that contain looped components, which are proven to operate correctly prior to integration but, composed into a reflective system will not necessarily operate as expected. This is expressed by the translation function of the serving component S and of the monitoring, respectively inspecting, component M , represented by the quotient $\frac{S}{1+SM}$. Dependability in reflective systems between the composed components of S and M is understood as the interference between the S - and the M - behaviour. In order to avoid non-deterministic behaviour like instability resulting from that behavioural interference, the a-priori analysis of the system dependability can provide with hints of how the serving and controlling components can be designed in order to avoid surprises from the composite architecture after deployment of the components.

3. Discussion

Because of the "self-reflecting", i.e. feedback loop build-into adapting systems a specific dependability between controlling and serving components exists. This dependability must be matters of verification prior to service set-up. Inspection without adaptation does not introduce the previously observed dependability because, there is no interference among components, e.g. of M and S . For the purpose of verifying global behaviour of dependable components one cannot rely on discrete interaction analysis. The notions of continuity and feedback by which components are bound into reflective system architecture enforce advanced design and verification methods that have to rely on behavioural characteristics like continuity, capacitivity and performance gradation. Adaptation is a continuous activity that is superpositioned to some service providing activities to be controlled. Methodologically, separation of concerns has been achieved by inventing Broy's continuous functions, that describe the behaviour of components as performance transformers that consumes flows and that produces flows continuously. A transformer is understood as a realistic component that needs resources like bandwidth, buffer space or computational power to achieve its individual functionality of e.g. transmission, storing or transformations etc. Transmission takes time, storing consumes buffers, and transformations may cause jitter or other distortions. This means, the input quality of a component will differ from the output quality of the same component. Thus, along the path through serving components the quality copes with a gradation. Distortions and gradations are considered as system influencing constraints. The ultimate goal of controlling is the smoothing of distortion and hence the adaptation of QoS gradation to some target values. It is highly demanded to use the analytical power of system theory for the verification of reflective systems which rely on continuous behaviour.

For our work on reflective systems (systems with feedback) at GMD we have started to realize parts of the auxiliary interfacing tools, mentioned above, that allow continuous inspections or activation of services under control. Further work is ongoing on the development of a reflection framework and on a notation by which reflection facilities can be generated and specified respectively. By the methodology, reflection facilities are allowed to be declared at any one of the components' interfaces. They support conformance checks during dynamic binding. The notational concepts have been presented in [Jdm98]. Early attempts to model reflective systems by separation of concerns have been presented in [Jdm96, Jdm97].

It seems, the notion of controlling is more restrictive than that one of reflection. Controlling is always oriented towards a target, whereas reflection provides with openness to any kind of modification as long as some given architectural or functional consistency constraints are not violated. The invention of the principles of control into the middleware could be said to be a first attempt to apply concrete methods and policies to a specific kind of reflection.

References

- [Blair98] L. Blair, G.S. Blair: The Impact of Aspect-Oriented Programming on Formal Methods (Position Paper); Proceedings of the ECOOP Aspect-oriented Programming Workshop Brussels 1998, Springer Verlag.
- [Broy93-a] M.Broy: Functional Specification of Time-Sensitive Communication Systems. ACM ToSEM Vol.2, No.1 1993 pp.1-46
- [Broy93-b] M.Broy, F.Dederichs, C.Dendorfer, M.Fuchs, T.F.Gritzner, R.Weber: The Design of Distributed Systems - An Introduction to FOCUS, TUM-19202-2, SFB Bericht Nr. 342/2-2/92A, Januar 1993
- [Broy84] M.Broy: Semantics for Communication Processes. Information and Control Journal Vol.6, No.3 1984 Academic Press
- [IS10746] ISO/IEC 10746 Part 1-3 (1995): IT Open Distributed Processing Framework
- [I363-1] ITU-T Recommendation I.363.1: B-ISDN ATM Adaptation Layer Specification, Type 1 AAL, Geneve 1996
- [Jdm96] Jan deMeer, GMD-FOKUS Berlin: Formal Handling of Streams in Multimedia Environments, 9th AMFADS - Workshop on Application of Formal Methods to System Development: Telecommunications, VLSI and Real-Time Computerized Control Systems, Montreal, pp.191-200, October 1996.
- [Jdm97] Jan deMeer, GMD FOKUS Berlin: On the Specification of End-to-End QoS Control, IWQoS97 Proceedings pp. 195 - 199, New York, May 1997
- [Jdm98] Jan de Meer, Axel Rennoch, GMD-FOKUS Berlin: Towards a QoS Binding Notation, in proceedings of the National Workshop on Formal Description Techniques FBT98 in Cottbus Germany, June 1998
- [LiNa98] B.Li, K.Nahrstedt: A Control Theoretical Model for Quality of Service Adaptations, 6th IWQoS Proceedings, Napa Valley California, May 1998.
- [Kuo95] B.C.Kuo: Automatic Control Systems, 7th edition, Prentice Hall, ISBN 0-13-304759-8
- [Sch93] H.Schlitt: Regelungstechnik -Physikalisch orientierte Darstellung fachübergreifender Prinzipien, 2nd edition. Vogelverlag Würzburg, ISBN 3-8023-1483-2