Adaptation in Software Radio using a Complex Organic Distributed Architecture (CODA)

Tereska Karran
email: Karrant@wmin.ac.uk
University of Westminster, London UK

ABSTRACT

CODA is an architecture sited within knowledge management, artificial intelligence and control systems paradigms. It defines components and interactions for the development of an intelligent knowledge management environment using complex stored data. A CODA subsystem was built for the CAST project. The CAST demonstration consists of three programs offering advice on the reconfiguration processes required to maintain quality of service under a range of environmental conditions. The sub-system was capable of self-adaptation under a variety of environmental conditions and was able to recognise crisis and recover sufficiently to maintain minimal services. The paper discusses results of the CAST demonstrators showing how data is sub-typed and how new rules and generalisations are added as it is transferred between layers. The CODA sub-system produced an integrated response in the event of multiple and unpredictable conditions in the environment while learning about more normal trends and improving response over time.

I. INTRODUCTION

CODA is based on a combination of Beer's cybernetic layered architecture [1] and knowledge discovery processes. Where possible layer 'memory' is organised using cognitive models as a guideline for the filtering a restructuring of data. CODA uses roles, layers and restructured data housed in warehouses, all of which ensure that cells only interact via filters and feedback mechanisms and that warehouses are secure and maintainable. It does this by using cellular structures to maintain data security and layers, and filters to ensure the data is in the right format. We focus on memory using concepts from cognitive psychology and from knowledge management systems with a view to providing systems intelligence [2]. A common feature of problems amenable to CODA-type applications is that external environmental data sources affect the efficiency of the system over time. Humans use intuitive pattern matching combined with data processing and some logic to solve problems. There seems to be no fixed structure to the processing, and it is at least partly experiential. Boden [3] describes this as 'global perceptual schematisation'. It requires some parallel processing as input data is acquired through a variety of channels. CODA manages a similar kind of schematisation in two ways, firstly by distributing the tasks involved in meeting objectives among cells, secondly, by organising cell clusters sharing tasks into subsystems. Figure 2 describes the identify-learn-adapt cycle used in CODA. CODA chooses between alternative paths while performing distributed tasks to meet objectives. Two heuristics are applied. Firstly, CODA analyses past success and failures in the history data and extrapolates rules. This is performed using layered views of the same core data. Secondly, CODA can apply a more open logic at the higher layers. This more open logic depends on CODA having a contextual history.

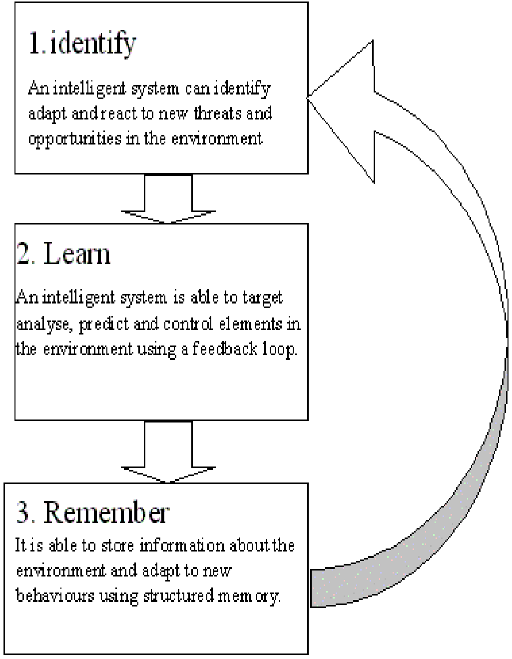


Figure 1. The CODA intelligence cycle.

CODA manages complex information by layering data and structuring it according to high-level objectives. Tasks are initiated by roles which express objectives in a formalised way. The cells execute tasks in order to meet objectives. Cells are usually clustered into substores data about their execution, it adapts and modifies its responses using an identify-learn-adapt cycle. The software used in the construction can be of any type, but to use organic structures and strategies in development. We start by building cells then add layers filters and the feedback mechanism [4].

II. CODA IN THE CAST PROJECT

CODA/CAST was part of the Information Societies Cluster of Projects on 'Reconfigurabilty' in mobile networks with a view to investigate and demonstrate new concepts and technologies in the third and fourth generation of mobile radio. The aim was to provide a standardised architecture made up from clearly defined mobile hardware and software components. CODA was used in the CAST project [5] to provide intelligent advice for the intelligent reconfiguration of the mobile network. Mobile Networks face the problem of delivering digital content at ever increasing bandwidth to greater numbers of users and at a reliable service quality [6]. Since this is likely to entail hardware components needing to reconfigure themselves for multiple circumstances, the provision of services could involve reconfiguration of the mobile station, base station and mobile switching station for effective service provision [7]. CODA software provided intelligent advice for the reconfiguration process as shown in the figure 2.

CODA operates within a socio-technical environment, where technologies and solutions evolve, and the 'social' user environment is changing and unstable [8] The rules and conditions defined for testing may be correct on formal scientific grounds and 'correct' in terms of formal specification. Yet a socio-technical system may fail for a multitude of other reasons related to environmental socio conditions. For example, the deployment of fourth generation mobile systems has been delayed by the poor user response to third generation systems. The reasons behind this are a combination of socio and technical factors leading to a mistrust of the new technology.

In the CODA demonstration we simplified the environment, although this weakens the validity of results, it helps to prove the principles involved, so there is a trade off.

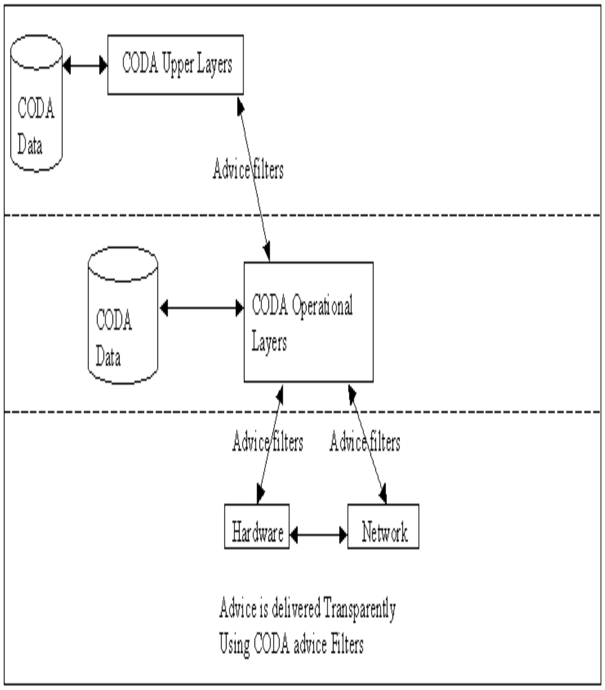
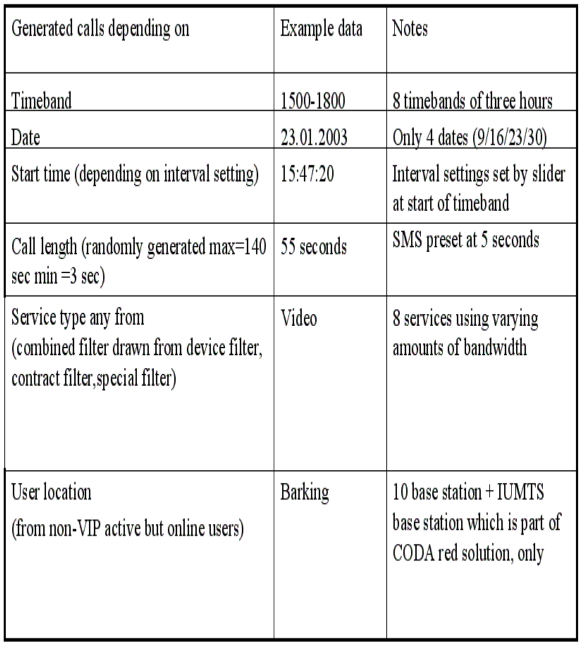


Figure 2. CODA in the CAST project

Thus our 'real' environment required the expression of time, group and location variables as a minimum. Environmental conditions were expressed as normal, threat or opportunity. Within this environment, variety in the group dimension, expressed as objects of interest, occurs as calls originate from various users with different types of mobile devices, contracts and service requests. Users are at 10 possible BSC locations. Calls occur over a timeband on a day-type, (the test environment only considers one non-holiday weekday). As calls are executed, the sub-system attempts to adapt to and modify the environment by either restricting or encouraging usage to make effective use of the lines and bandwidth available by means of special filters. Adjustments are based on assessment of operating. It is assumed that contracts are based on analysis of past bandwidth usage and on predicted future usage as well as business considerations, using calculations based on normal conditions. Since complex non-linear systems do not always behave predictably we included both opportunity and threat conditions in our artificial environment.

Specially designed call generators were used. The resulting test data had few clusters compared to 'real' data. However, we have foreknowledge of the patterns in the data, and it is possible to test if CODA successfully identifies them and produces reasonable responses.



Generated calls depending on	Example data	Notes
Timeband	1500-1800	8 timebands of three hours
Date	23.01.2003	Only 4 dates (9/16/23/30)
Start time (depending on interval setting)	15:47:20	Interval settings set by slider at start of timeband
Call length (randomly generated max=140 sec min =3 sec)	55 seconds	SMS preset at 5 seconds
Service type any from (combined filter drawn from device filter, contract filter,special filter)	Video	8 services using varying amounts of bandwidth
User location (from non-VIP active but online users)	Barking	10 base station + IUMTS base station which is part of CODA red solution, only

Figure 3. Producing call data for CAST

Producing realistic data proved difficult. Filters allowed the generator to pivot through data (see figure 3). Although normal conditions appear fairly random, there are many clusters in real data. It was necessary to produce environmental data demonstrating different call patterns, so as to model behaviour within normal, threat and opportunity ranges.

III. DEMONSTRATION

The sub-system design aimed to potentially co-operate with several other interacting sub-systems integrated by 'central processing' layers. Testing for the CAST sub-system concentrates on responsiveness to call patterns based on the identify learn and adapt cycle.

Identify: The system uses the feedback mechanism to identify environmental conditions, normal threat or opportunity, in order to apply the right type of filters.

This is demonstrated by reconfiguring silver users to UMTS in the case of BSC failure.

Learn: Under normal operating conditions the system should be able to predict the correct filter settings for effective provision at each level of the system.

This is demonstrated by showing how users are provided with filter settings depending on time band and device. The Base Station is able to adjust itself depending on the conditions of the bandwidth and the number of lines.

Adapt: Semi autonomous operation is possible because suitable operating ranges are defined in the critical success factors. Calls on the higher layers should only be made if

This is demonstrated by showing how special filters could be adapted to manage different environmental conditions over time.

IV. DESIGN

In CAST, reconfiguration service requirements evolve depending on environmental and operating conditions. Provision of services may entail reconfiguring the mobile station, base station and mobile switching controller. The CODA cell mechanism means that the advice system can support third and fourth generation systems while continuing to support existing second generation systems [Madani et al (2002)]. In order to do this, CODA models network usage, based on call patterns from history data. CODA cells at layer 3 predict new usage patterns and the products of their analysis are fed back to adjust management of multiple distributed network components using critical success factors. Adaptability hinges on 'memory' managed across several warehouses and outputs decisions about the reconfiguration of all components in the network. Such decisions are important for the successful delivery and deployment of third and fourth generation networks in the large scales planned over the next few years. The European long-term plan is to move all possible communications into software radio, therefore to ensure confidence in air interface networks operational reconfiguration needs to be fast and efficient.

The problem is that the network needs to ensure that services are met while maintaining balanced usage. However, some users may be logged on to mobile devices which do not support their contract options while others may be have device capability to leverage more services than their contract allows, still others are not active or failing to make full use of device capability. This can lead to fluctuations in call usage. It may leave the network under-used. The sub-system adjusts special filters depending on predicted and actual system. At the same time, the network needs to adjust to possible non-linear crisis conditions, where all users try to leverage services unexpectedly. Each cell has its own special filters. Under normal conditions they should be the same, with local adjustments depending on environmental conditions. To maintain control, each cell contains dynamic variables which help the system to locate and correct faults and improve performance. Cell filter settings depend on its internal critical success factors and these in turn, depend on its internal operating status. Filter adjustments are mechanistic and intelligence for the settings is managed by the upper layers. In the case of crisis, the cell imposes pre-set crisis filters, and notify the higher layer that it is in crisis[9]

V. IMPLEMENTATION

Three test programs demonstrate different elements of the sub-system. Two programs demonstrate the first stage of adaptation within monitoring operations cells. The third program, involves prediction on the basis of history data and uses several sets of results under different conditions over time. We have concentrated on showing the modification of the special filters since an automated modification procedure best meets the sub-system objective, which is to improve quality of service. However it is possible to add other objectives. For example, new clusters may modify data by adding new sub-types such as ‘emergency’ or ‘gold 24carat’ users, or they add new rules such as ‘suggest add new bandwidth to BSC’ which is regularly flagging a yellow condition.

CODA Blue shows how the cell-cluster meets the objective to ‘maintain balanced usage of lines and bandwidth.’ Critical success factors describe three possible environmental states and adjustment to the filters is under cell control. Firstly, authentication shows the activation of a user cell and the initialisation and setting of the filters. Secondly, the demonstrator shows how BSC cells manage multiple calls using preset filters and critical success factors. It is possible to view the status log and the call log as each is being built up on the screen The program meets objectives by diverting to other base stations in the event of a shortage of bandwidth, reconfiguring by adjusting filters to downgrade service, and by requesting advice from higher layers in the event of a failure to meet contractual service.

CODA Red demonstrates how cells support dynamic reconfiguration within a sub-system without affecting the overall performance of other elements in the network. The objective is expressed as ‘maintain minimum service under failure conditions.’ Gold contracts have access to UMTS. In the event of a GSM base station saturation, the sub-system checks the UIR for users logged on to the failing base station [10]. Silver contract users are online, located at the failed base station and trying to make a call, the user handsets with UTRA capability will be reconfigured for UMTS. The objective is effected by adding a rule to the operating conditions of all base stations and the MSC as follows:

If a base station goes into condition red and all other base stations are in condition yellow or red, then identify all active and online silver users, with device capability, located at red base stations and reconfigure them to use the wide-area UTRA base station.

The critical success factors describe the indicators of system failure and the adjustment possible is to the filters under cell control. It is possible to identify more conditions using a cell-cluster and layered decision structure using simple programming procedures.

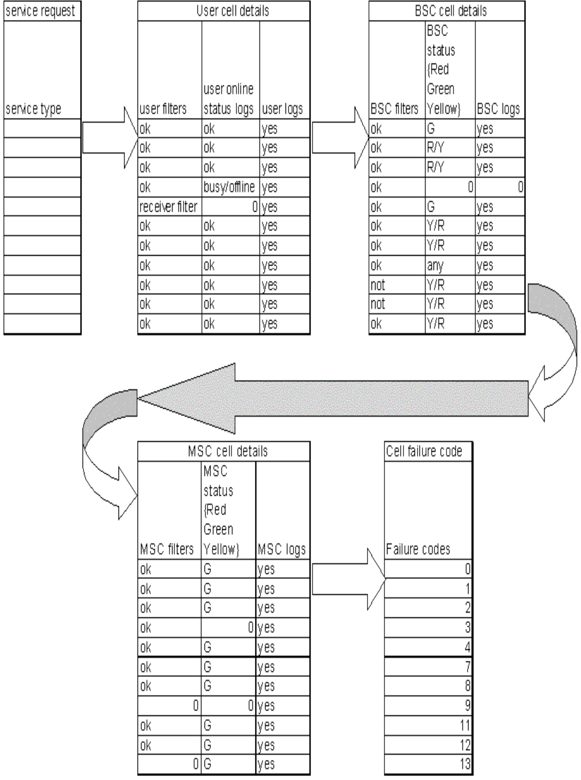
CODA Green contains 10 functioning Base Stations as opposed to one, and a functional MSC cell. Cells at all 3 layers interact to improve quality of service. Input information is in the form of cell performance logs from the monitoring operations layer and adjustments are in the form of special filters for a timeband depending on operating conditions. The system improves its responses, but as it reaches stasis, it is in danger of stagnation. As systems reach technical stasis, social pressure to change make the system ‘out-dated’ and new technology is generated. The socio-technical change-process for mobile systems involves three new generations in ten years and the pace is continuing. Thus, we assume modifications at any time.

Failure type	
code	Code description
0	Call ok
1	1 divert used
2	2 diverts used
3	Receiver busy
4	Receiver cannot take
5	Mobile device failure (caller)
6	Mobile device failure (receiver)
7	Receiver BSC rejects non-contract call (BSC yellow/red)
8	Start BSC rejects non contract call (BSC yellow/red)
9	Start BSC filters out a non contract call
10	MSC filters out non contract call (not in demonstrator)
11	Receiver BSC fails to divert contract call
12	Caller BSC can’t divert a contract call
13	Caller BSC filters out a contract call

Figure 4 Types of failure detected by CODA

In normal operation, layer 3 cells find failure clusters and compares actual failures to predictions at the end of each timeband. Just as in biological homeostats, failures are of immediate interest and some failures are more critical than others. Figure 4 shows the range of failures identified by the CODA demonstrator. The range can be redefined on the basis of data

A service may ‘fail’ to execute at several stages of the call cycle. Each cell has its own status log, call logs and filters. If a failure occurs in a cell, the details are logged in its call log and status log. Cells work co-operatively, analysing the core data, looking for unusual clusters such as failure by a particular device, or by a base station, or by a timeband. Discrepancies in current operations are analysed and it is possible that the special filters will be adjusted or new rules will be added. For example, the ‘settings cell’ checks the current status logs and checks users are behaving as predicted while the ‘comparison cell’ checks the previous timeband and looks for new clusters and structures outside predictions. Figure 5 shows adaptation.



service request
service type

User cell details		
user filters	user online status logs	user logs
ok	ok	yes
ok	ok	yes
ok	ok	yes
ok	busy/offline	yes
receiver filter	0	yes
ok	ok	yes
ok	ok	yes
ok	ok	yes
ok	ok	yes
ok	ok	yes
ok	ok	yes

BSC cell details		
BSC filters	BSC status {Red Green Yellow}	BSC logs
ok	G	yes
ok	R/Y	yes
ok	R/Y	yes
ok	0	0
ok	G	yes
ok	Y/R	yes
ok	Y/R	yes
ok	any	yes
not	Y/R	yes
not	Y/R	yes
ok	Y/R	yes

MSC cell details		
MSC filters	MSC status {Red Green Yellow}	MSC logs
ok	G	yes
ok	G	yes
ok	G	yes
ok	0	yes
ok	G	yes
ok	G	yes
ok	G	yes
0	0	yes
ok	G	yes
ok	G	yes
0	G	yes

Cell failure code
Failure codes
0
1
2
3
4
7
8
9
11
12
13

Figure 5 CODA failure identification

The ‘normal’ cycle for adaptation is summarised in figure 6 below. At the end of each timeband, performance data is taken from monitoring operations cells and merged into a ‘layer 3 warehouse’ containing all the four weeks of call data. The process has been simplified in our demonstration, so the warehouse is a large cube-type spreadsheet. The quantity of data is very large, even though the monitoring operations layer has already ‘filtered’ out unnecessary detail. Filtration between layer 2 and layer 3 warehouses means that data unnecessary to this particular sub-system (not to other sub-systems) is ‘forgotten’. The data is added to, week, by week, so that the monitoring cells get a clearer view of trend behaviour as more data is acquired. In this demonstration the data selection process has been highly simplified as we are concentrating on a proof of adaptation. However, the selection of data using filters is an important part of the intelligence cycle, otherwise the system would be swamped by unmanageable amounts of data. The core filtered data is analysed at the end of each timeband, with a view to improving performance by adjusting special filters for the next offering of that particular timeband. The data is analysed for trends, clusters and unusual behaviour patterns.

VI. RESULTS

It proved possible to adjust and fine-tune the filters more finely than anticipated. Filters were adjusted for a timeband, but it is possible to adjust them more finely in future applications, particularly during a crisis. The decision-making process is widely distributed, so that it is possible to make fine distinctions without affecting the rest of the system detrimentally. We based our system crisis on a real scenario, where multiple base stations were unable to meet services on January 30th 2003. Initial testing of a similar scenario on the CODA/CAST sub-system loaded with artificially created data, suggests that a CODA sub-system would be able to maintain services, although most users would be limited to voice or even to sms. Although the crisis would affect quality of service, we believe that maintaining a limited service would prevent loss of confidence in the reliability of a mobile network on the part of the users. This would have to be confirmed by another sub-system which used different data to measure confidence in the network.

VII. CONCLUSIONS AND FUTURE WORK

CODA is able to store data efficiently by restructuring data and retaining the minimum in the operational memory. The timeband approach allows system designers to decide on the optimum chronon for the efficient operation while still allowing higher layers to observe patterns and trends in the data. Subtyping and ranging key variables allows us to create dynamic structures as the data is transferred between layers. We have assumed that information normally required from higher layers is provided by a manual operator, because full decision capacity requires advice from higher layers. For example, the system needs to decide if a usage cluster of sms messages signifies a new trend, or because there is a football match, or because a television programme has requested for voting from a young audience.

In the scenario run, BSCs imposed the filters without any problem and the problem was localised within specific areas without affecting the rest of the system in south London base stations.

Crisis does not affect trend analysis as 'non-linear data' is marked as generated under crisis, so the system could easily ignore it and return to filters for the previous week, but some decision have to be made as to whether the imposed filters actually worked and whether they were perhaps too stringent, and applied too quickly. There needs to be a crisis analysis cell in later research.

It seems that the adaptive response from CODA can be quite fine-grained despite the large numbers of calls and the complexity of the response required. This is due to the distributed nature of the decision process, which supports adjustments by cells, cell types or under devolved control. Decisions behind the reconfiguration process are transparent to the reconfiguration modules and can be delivered in any format using the CODA filters, so it should be possible to add CODA to existing mobile networks. The system succeeds in its main purpose of keeping the network stable and providing services even in adverse conditions. The main weakness of the test process is the artificiality of data. This meant that CODA was unable to reveal any interesting emergent trends because of the relatively small size of system, the small number of users, and the number of resultant calls.

Adding a control layer is unlikely to involve radical new software solutions. This is because CODA is primarily an architecture managing multiple information systems components and disparate information from distributed sources and not a programming tool. The control layer makes use of existing solutions, and will not affect the performance of existing sub-systems detrimentally. Control layer cells improve and co-ordinate sub-systems without operator intervention. Performance data from the sub-systems needs to be organised using suitable structures to support semantic and episodic memory. The logical next step is to build more subsystems and develop the next co-ordinating and integrating control layer. For example, the long-term vision for mobile services is to introduce fourth generation systems delivering a wide range of digital services, but the up-take of third generation mobile services has been hampered by user disaffection over poor quality of service. The sub-system we have constructed is only one part of a mobile 'vision' for delivering services and the sub-system has to be integrated with other sub-systems managing customer relationships and detecting usage trends. These additional sub-systems need to be integrated.

Acknowledgement

This document is based on work carried out in the EU sponsored collaborative research project CAST. Nonetheless only the authors are responsible for the views expressed here

REFERENCES

1. **Espejo**, R., Schuhsmann, W. & Schwaninger, M. (1996) *Organizational Transformation and Learning, Cybernetic Approach to Management.* Wiley.
2. **Karran**, T. & Justo, G.R. (2002) Architecture Adaptation and Reconfiguration using Critical Success Factors *PDPA Conference* Las Vegas June.
3. **Boden**, M. (1987) Artificial Intelligence. In R.Gregory (Ed) *Oxford Companion to the Mind*. Oxford: Oxford University Press.
4. **Karran**, T., Ribeiro-Justo, G. & Zemerly, J. (2000) An Organic Architecture for Component Evolution in Decision Systems, 4th World Conference on Systemics, *Cybernetics and Informatics (SCI 2000),* Orlando, USA, July, IIIS.
5. **Patel**, P. Madani, K., Karran, T., & Lohi, M. (2000) Configurable radio with Advanced Software Technology (CAST) – Initial Concepts" IST Mobile Communications Summit 2000, October 1-4, 2000, Galway, Ireland, pp. 139-144.
6. **EU** Vision on Mobile and Wireless Telecommunications (2002)Andersen
7. **Buracchini**, E (2000) The Software Radio Concept. *IEEE Communications Magazine*, September 2000, pp.138-143.
8. **Bertalanffy**, L.V. (1981) General Systems Theory – A Critical Review. In Systems Behaviour. Open University Press.
9. **Karran** T., Madani, K., Justo, G.R., (2003) Self Learning and Adaptive Systems. In Dillinger, M. (Ed) *Software Defined Radio.* Wiley.
10. **Lund,** D. Honorary B, Madani K, Characterising Software of the Physical Reconfigurable Radio Subsystem. Proceedings of IST2001, Barcelona, September 2001, pp357-362.