
Abstract

Software radios are emerging as platforms for multiband multimode personal communications systems. Radio etiquette is the set of RF bands, air interfaces, protocols, and spatial and temporal patterns that moderate the use of the radio spectrum. Cognitive radio extends the software radio with radio-domain model-based reasoning about such etiquettes. Cognitive radio enhances the flexibility of personal services through a Radio Knowledge Representation Language. This language represents knowledge of radio etiquette, devices, software modules, propagation, networks, user needs, and application scenarios in a way that supports automated reasoning about the needs of the user. This empowers software radios to conduct expressive negotiations among peers about the use of radio spectrum across fluents of space, time, and user context. With RKRL, cognitive radio agents may actively manipulate the protocol stack to adapt known etiquettes to better satisfy the user's needs. This transforms radio nodes from blind executors of predefined protocols to radio-domain-aware intelligent agents that search out ways to deliver the services the user wants even if that user does not know how to obtain them. Software radio [1] provides an ideal platform for the realization of cognitive radio.

Cognitive Radio: Making Software Radios More Personal

Joseph Mitola III and Gerald Q. Maguire, Jr.
Royal Institute of Technology

A Global System for Mobile Communications (GSM) radio's equalizer taps reflect the channel multipath structure. A network might want to ask a handset, "How many distinguishable multipath components are you seeing?" Knowledge of the internal states of the equalizer could be useful because in some reception areas, there may be little or no multipath and 20 dB of extra signal-to-noise ratio (SNR). Software radio processing capacity is wasted running a computationally intensive equalizer algorithm when no equalizer is necessary. That processing capacity could be diverted to better use, or part of the processor might be put to sleep, saving battery life. In addition, the radio and network could agree to put data bits in the superfluous embedded training sequence, enhancing the payload data rate accordingly.¹

Two problems arise. First, the network has no standard language with which to pose a question about equalizer taps. Second, the handset has the answer in the time-domain structure of its equalizer taps, but cannot access this information. It has no computational description of its own structure. Thus, it does not "know what it knows." Standards-setting bodies have been gradually making such internal data available to networks through specific air interfaces, as the needs of the technology dictate. This labor-intensive process takes years to accomplish. Radio Knowledge Representation Language (RKRL), on the other hand, provides a standard language within which such unanticipated data exchanges can be defined dynamically. Why might the need for such unanticipated exchanges arise? Debugging new software radio downloads might require access to internal software parameters. Creating personal services that differentiate one service provider from another might be enhanced if the provider does not need to expose new ideas to the competition in the standards-setting process. And the time to deploy those personalized services could be reduced.

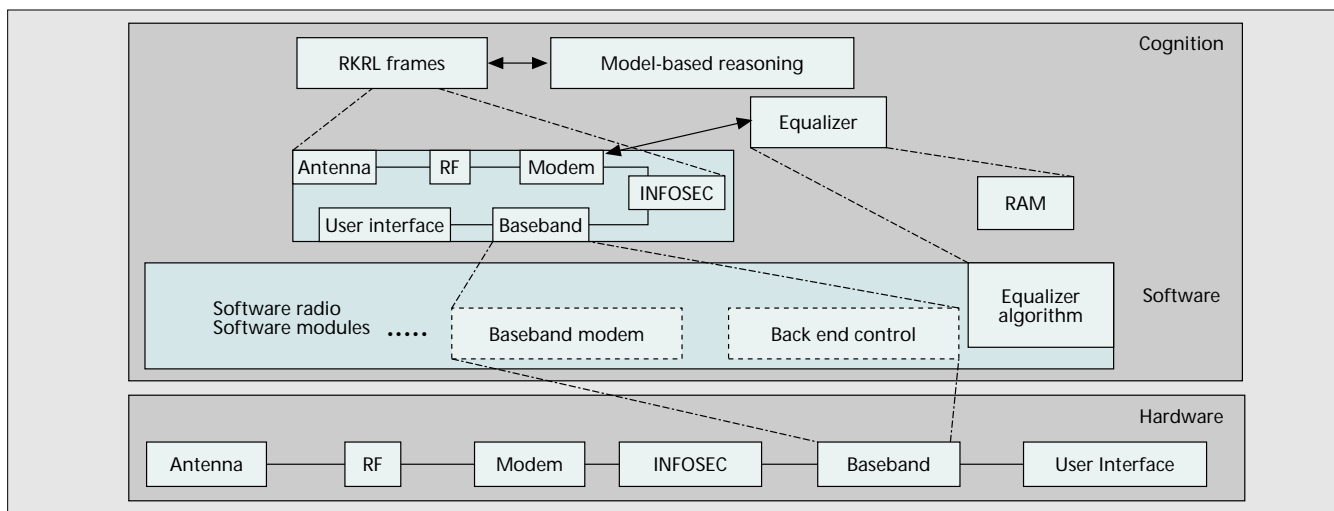
Cognitive radio, through RKRL, knows that the natural lan-

guage phrase *equalizer taps* refers to specific parameters of a tapped delay-line structure. This structure may be implemented in an application-specific integrated circuit (ASIC), a field programmable gate array (FPGA), or an algorithm in a software radio. Since a cognitive radio has a model of its own internal structure, it can check the model to find out how the equalizer has been implemented. It then may retrieve the register values from the ASIC (e.g., using a JTAG port) or find the taps in the proper memory location of its software implementation. A radio that knows its own internal structure to this degree does not have to wait for a consortium, forum, or standards body to define a level H33492.x7 radio as one that can access its equalizer taps. The network can pose such an unanticipated question in (a standard) RKRL, and *any* RKRL-capable radio can answer it. To enable such a scenario, cognitive radio has an RKRL model of itself that includes the equalizer's structure and function, as illustrated in Fig. 1.

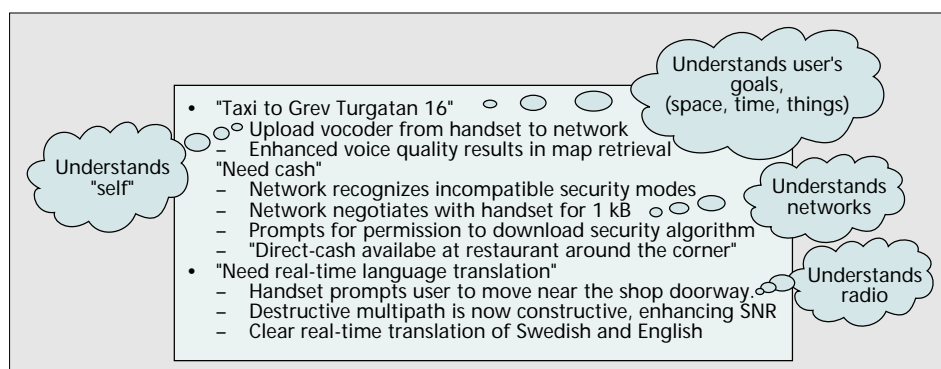
In this example, the radio hardware consists of the antenna, the radio frequency (RF) conversion module, the modem, and the other modules shown in the hardware part of the figure. The baseband processor includes a baseband modem and a back-end control protocol stack. In addition, this processor contains a cognition engine and a set of computational models. The models consist of RKRL frames that describe the radio itself, including the equalizer, in the context of a comprehensive ontology, also written in RKRL. Using this ontology, the radio can track the user's environment over time and space. Cognitive radio, then, matches its internal models to external observations to understand what it means to commute to and from work, take a business trip to Europe, go on vacation, and so on.

Clearly, significant memory, computational resources, and communications bandwidth are needed for cognitive radio, so this technology might not be deployable for some time. In addition, a collection of cognitive radios may not require human intervention to develop their own protocols. Initially, intervention will be required in order to ensure that networks of such radios remain stable (or that we know who to blame if this is not the case). Networks of such radios are complex

¹ This raises a host of questions about the control of such complex adaptive agents, network stability, and the like.



■ Figure 1. Cognitive radio.



■ Figure 2. More personalized service concepts.

adaptive systems [2], the study of which is an emerging discipline concerned with the nonlinear behavior of large collections of adaptive entities that have complex interactions. Although there are many technical challenges, the opportunities for enhanced personal services motivate the development of cognitive radio. This article, therefore, outlines the key technical ideas behind cognitive radio, RKRL, and related research at the Royal Institute of Technology (KTH), Sweden.

Personalized Services Scenarios

The services enhancements to be enabled by cognitive radio are motivated by a set of use cases [3] that require the radio to have an advanced degree of “understanding” of topics illustrated in Fig. 2. Next-generation personal communications services (PCS) will know the location of handsets and wireless personal digital assistants (PDAs) to within 125 m for emergency location reporting. Location-aware research [4] is creating technologies for location-aware services, such as flexible directory services [5]. Cognitive radio adds locally sensed recognition of common objects, events, and local RF context. Thus, for example, a cognitive radio can infer the radio-related implications of a request for a taxi to a specific address. It can then tell the network its plan to move from its present location to “Grev Turgatan 16.” The network then knows that this user (with high probability) will move across three cell sites into a fourth within the next ten minutes. If this user is headed for a conference center equipped with a local cell-phone jammer, it is unlikely to offer the usual load to the network after the taxi ride. Such exchanges could reduce uncertainty about the load offered to a network, potentially

enhancing the efficiency of the use of radio resources.

Software radios as presently conceived cannot have such an intelligent conversation with a network because they have no model-based reasoning or planning capability and no language in which to express these things. For example, a software radio from the United States may have the RF access, memory, and processing resources to operate in Sweden. If it lacks compatibility with release level G of the host service provider, it will not work. A soft-

ware radio cannot “discuss” its internal structure with the network to discover that it can be reconfigured to accept a download of the required software personality. Cognitive radio, however, employs a rich set of internal models useful for a wide range of such dialogs. In addition, the space-time models of the user, network, radio resources, and services personalize and enhance the consumer’s experience. The analysis of such use cases yielded a large set of models, conceptual primitives, and reasoning schema necessary for cognitive radio. Which computer languages should be used to express these things?

Radio-Related Languages

In addition to natural language, several computer-based languages are relevant to the expression of radio knowledge (Table 1). The International Telecommunication Union (ITU), for example, adopted the Specification and Description Language (SDL) in its Z.100 Recommendations. SDL readily expresses radio state machines, message sequence charts, and related data dictionaries. The European Telecommunications Standards Institute (ETSI) recently adopted SDL as the normative expression of radio protocols, so one expects SDL modeling of radio to continue to expand. SDL, however, lacks primitives for general ontological knowledge needed, for example, to reason about a travel itinerary.

The Unified Modeling Language (UML) resulted from the unification of diverse object-oriented analysis, modeling, design, and delivery methods. This language readily expresses software objects, including attached procedures (*methods*), use cases, and the packaging of software for delivery. In principle, it can be used to model common-sense knowledge, including

Language	Strengths:	Lacks (or not designed for)
SDL [6]	State machines, message sequence charts, large user base, much encoded knowledge	Plan representation, uncertainty
UML [7]	General ontologies, structure, relationships	Hardware, RF propagation
IDL [8]	Interfaces, object encapsulation	General computing
HDLs [9]	Hardware, electronic devices, interfaces	Geospatial, plans
KQML [10]	General dialogs (ask/tell), tagged semantics	General computing
KIF [11]	Axiomatic treatment of sets, relations, numbers, frames; ontologies	General computing, hardware, RF propagation

■ **Table 1.** *Radio knowledge languages.*

```
(Tell :language RKRL :ontology Stockholm/Europe/Global/Universe/Version 0.1
:Move_Plan (:owner User (:from Kiosk :to "Grev Turgatan 16") :distance 3522 m
:via (Taxi :probability .9) (Foot :probability 0.03))
(:PCS-needs (:DECT 32kbps) (:GSM GPRS) (:backlog Composing-email)))
```

■ **Figure 3.** *The KQML expression of a plan.*

plans, space, time, relationships, people — just about anything. In practice, it has a strong presence in software design and development, but is weak in the modeling of hardware devices. In addition, although UML can provide a design framework for radio propagation modeling, the target languages are likely to be C or FORTRAN for computational efficiency in tracing tens of thousands of rays of radio waves.

The Common Object Request Broker Architecture (CORBA) defines an Interface Definition Language (IDL) as an implementation-independent syntax for describing object encapsulations. In addition to the 700 companies that comprise the Object Management Group (OMG), IDL is being used by the Software-Defined Radio (SDR) Forum [12] to represent interfaces among the internal components of SDRs. Since this language is specifically designed to declare encapsulations, it lacks the computational power of general languages such as C or Java. IDL excels at architecture integration (e.g., the interface to an equalizer ASIC), but not at expressing the functions and contributions of a component (e.g., the enhancement of bit error rate, BER, at low SNR).

The hardware description languages (HDLs), primarily Verilog HDL and VHDL, readily express the internal structure of ASICs and the personalities of FPGAs. However, cognitive radio does not need the level of detail present in most HDL data sets. Moreover, it needs to know the functions and contributions so that it can make trade-offs, create plans, and reprogram itself. While the documentation package associated with HDL may provide some of this insight, the information is not in a computationally accessible form.

The Knowledge Query and Manipulation Language (KQML), on the other hand, was explicitly designed to facilitate the exchange of such knowledge. Based on *performatives* such as *tell* and *ask*, KQML readily expresses the dialog about equalizer taps and multipath by introducing a few new *tags*. The KQML plan to take a taxi from the information kiosk to Grev Turgatan 16 uses the tell performative to tell the network of the plan, as shown in Fig. 3. In this example, the radio also warns the network that its user is composing some e-mail and so will need either a Digital European Cordless

Telecommunications (DECT) data channel or the GSM general packet radio service (GPRS) while in transit.

The ontology performative could invoke an existing ontology or could express the local context, as in this case. It normally would be defaulted unless it changed. The other declarations are self evident, which is one of the strengths of KQML. Like IDL, however, KQML is an interface language. Although, for example, one can express rules from a knowledge base using KQML, one must translate these rules into a convenient internal form (e.g., LISP or PROLOG)

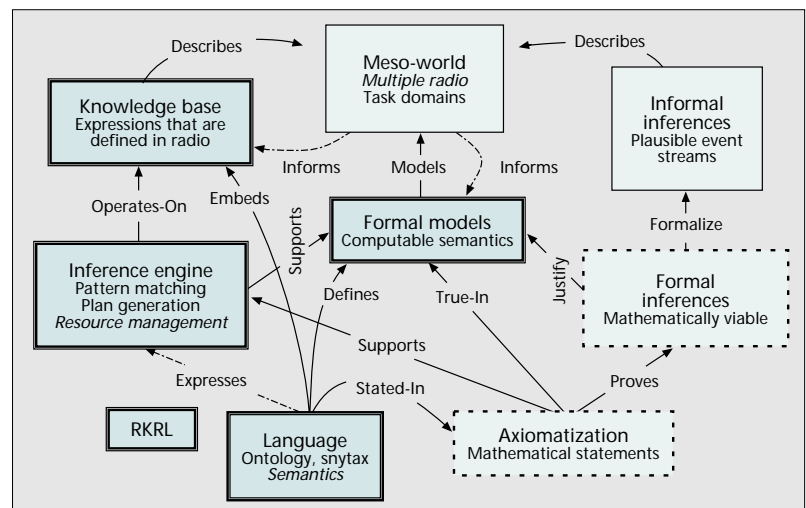
in order to use them. In addition, the expression of general spatial knowledge, such as the three-dimensional structure of adjacent city blocks, is better expressed in structured arrays than in KQML. KQML could be used to send changes to such arrays, however.

The knowledge interchange format (KIF) provides an axiomatic framework for general knowledge including sets, relations, time-dependent quantities, units, simple geometry, and other domain-independent concepts. Its main contribution is strong axiomatization. It has a LISP-like structure and, like IDL and KQML, is not specifically designed for internal use, like C or Java.

Finally, most radio knowledge is represented in natural language. It lacks precision, but in some sense has ultimate expressive power, particularly if one includes graphics and multimedia in natural language. Natural language suffers from ambiguities and complexity that at present limit its use as a formal language. RKRL v. 0.1 was created to fill the voids in the expressive power of computer languages while enforcing a modicum of structure on the use of natural language.

Cognitive Radio as a Chess Game

RKRL is supposed to represent the domain of information services that use software radios for mobile connectivity. Since a software radio has a choice of RF bands, air interfaces, data protocols, and prices to be paid, in competition with other users, the



■ **Figure 4.** *Meso-world structure of cognitive radio.*

Handle	Model	Body	Context
Global plane	Contains	South America	Physical world model/universe/RKRL 0.1

■ **Table 2.** An RKRL frame asserting that the global plane contains South America.

Handle	Model	Body	Context
Contains	<Process>	SetAccumulate	Models/Universe/RKRL/Version 0.1
SetAccumulate	Excel	<Control> <Shift>S	Contains/Models/Universe/RKRL/Version 0.1
Contains	<Test>	modelVar = "contains"	Models/Universe/RKRL/Version 0.1
Contains	<Domain>	Sets	Models/Universe/RKRL/Version 0.1
Contains	<Range>	Sets	Models/Universe/RKRL/Version 0.1
Contains	Definition	This string-model asserts that the body is a member of the handle.	Models/Universe/RKRL/Version 0.1

■ **Table 3.** Statements defining the verb contains as a formal model.

domain is analogous to a chess game. The network may orchestrate the game, or in some bands (e.g., the U.S. industrial, scientific, and medical band) cognitive radios will simply compete with each other, hopefully with some radio etiquette. The game board is the radio spectrum with a variety of RF bands, air interfaces, smart antenna patterns, temporal patterns, and spatial locations of infrastructure and mobiles. RKRL provides a consistent way of describing this game board. The future wireless PDAs are the game pieces. Which moves are legal? How will one move impact others in the neighborhood? If a game piece expresses its future needs or plan for use of services to the network, can the network better orchestrate the use of radio resources? And to what degree should the way in which spectrum is used over physical space, code space, power, parameter space, and time be defined by the mobile units themselves? No one knows the answers to these questions because software radios are just emerging, but the need to address them in the future seems clear. RKRL expresses the game board and the legal moves.

Davis [13] defines *micro-worlds* as performance domains for naive physics. Cognitive radio consists of the multiple micro-worlds (the *meso-world*) represented in Fig. 4.

The meso-world of RKRL 0.1 consists of the 41 micro-worlds summarized in Fig. 5. Each is structured according to formal models, and described in a knowledge base. Competence comes from the pattern matching and plan generation capabilities of a cognition cycle, mediated by the related inference engines. RKRL includes syntax and ontological information. Parsing an RKRL statement includes interpreting that statement in terms of the RKRL radio ontology and knowledge base. Thus, words, including KQML tags, have a meaning that is fixed in a given context (although a single word can have different meanings in different contexts). Thus, the scope of RKRL includes the formal models, knowledge base, inference engine, multiple syntaxes, and a radio ontology.

The expression of syntax in RKRL permits one to embed knowledge from external representations (especially SDL and UML). RKRL may now be described syntactically.

RKRL

Instead of attempting to replace SDL, UML, IDL, or KQML, RKRL integrates them through model-based

reasoning as follows. RKRL is a parallel frame language. Each RKRL statement is a frame:

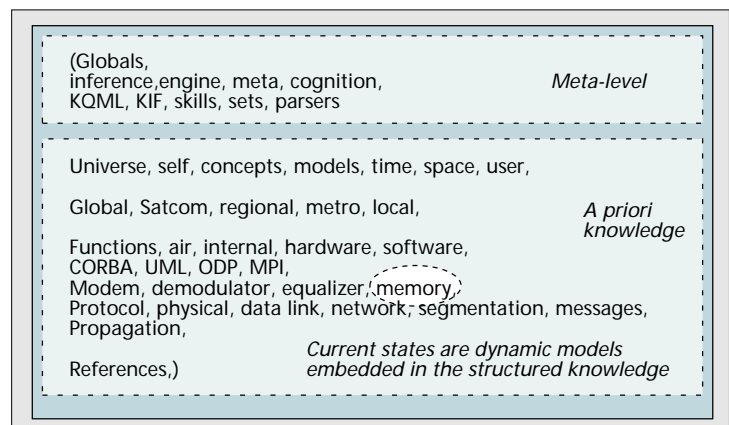
< frame > = [< handle >, < model >, < body >, < context >].

The frame expresses a relationship between the handle and body, in a given context. The < model > part defines the exact relationship being expressed. Handles should be thought of as names for things. If a thing contains other things, it can be viewed as an object. If not, it is a terminal constant. Frames are interpreted in parallel, like the cells in a spreadsheet. For example, the following RKRL statement says that South America is part of the global plane in the physical world model of the universe of RKRL 0.1. Additional frames assert Europe and the rest into the global plane. Since the global plane is part of the universe, it can be thought of as an

attribute of universe, with a value that is a set of regions of the world. However, it can also be thought of as just a list of the names of countries. These semantics (object, list, etc.) are *not* part of the semantics of RKRL. Instead, the semantics are explicitly declared and coded in RKRL using formal computational models (Table 2).

For example, the word *contains* is a verb from natural language, used in an obvious way. But contains is also a formal model defined in the model's micro-world using the statements in Table 3.

Contains, the model, is defined in terms of a < domain > (the micro-world of sets), a < range >, a < test > for membership, and a < process > for finding members in a local context. The first frame says that Contains has a < process > called SetAccumulate. The fourth line says that Contains is defined over sets (which is a complete micro-world). The items in < brackets > are defined in the meta micro-world. The < test > process consists of a chunk of Excel Visual Basic for Applications (VBA) macro language that will test a frame with a standard binding (modelVar is bound to the model part of the frame). These string models tell the RKRL interpreter (embryonic at present) how to use chunks of code to construct programs that create and manipulate objects, perform model-based reasoning, and otherwise control the software radio platform. To attach an existing VBA macro to an entity, one



■ **Figure 5.** RKRL micro-worlds.

states (line 2, Table 3) that its Excel model is <Control> <Shift>S, the associated keyboard macro. In addition, one may redefine contains in some other context. If it is undefined in a local context, the interpreter searches the micro-worlds ascending the context hierarchy, and then looks across micro-worlds until it either finds a definition or sets a goal of getting a definition. In RKRL, to Get includes to Create (through inferencing) and to Inquire (e.g., of the user or network).

Knowledge about Equalizers

For a more radio-oriented example, consider the adaptive equalizer. RKRL knows about the equalizer running on its own platform from the following context: Equalizer/Demodulator/Modem/Internal/Fine Scale/Immediate/Local/Metropolitan/Regional Plane/Global Plane/Physical World Model/ Universe/RKRL/Version 0.1. It also knows about a generic equalizer in Equalizer/Demodulator/Modem/Concepts/RKRL/Version 0.1. Facts that are known a priori of an equalizer include a frame [Equalizer, Property, Reduces intersymbol interference ...]. The natural language phrase *reduces intersymbol interference* (ISI) is parsed because the concept of reduce has the Excel fragment "handleVar(bodyVar)<bodyVar" and ISI is defined in the physical layer as intersymbol interference with the property that it increases BER. Since all the elements reduce to either Excel, VBA, or a call to an embedded model, the frame is completely interpretable.

The equalizer is defined from the <domain> IF-signal onto the <range> IF-Signal. Its taps are defined using statements such as [Tap-0, Numerical model, 1.2745, Taps/Delay Line/Equalizer/...]. The output is defined in a frame [Weighted-result, Numerical model, *, ...], where * is the Excel expression:

$$\begin{aligned} &= \text{Weight 3 Model} * \text{Tap 3 Model} \\ &+ \text{Weight 2 Model} * \text{Tap 2 Model} \\ &+ \text{Weight 1 Model} * \text{Tap 1 Model} \end{aligned}$$

If the present RKRL were embedded in a PDA, the a priori model of a three-tap equalizer would be as above, but the internal model would be a dynamic model. Dynamic models contain the values from the current system that they are mod-

	Plane	Objects	Space	Time	Information sources
1	Global	Regions	10,000 km	1 yr	Travel itinerary
2	Regional	Cities	1000 km	1 week	Weekly planner
3	Metropolitan	Districts	100 km	1 day	Commuting pattern
4	Local	Buildings	1 km	1 hr	GPS, lunch?
5	Immediate	Rooms	100 m	1 s–1 min	Dead reckoning
6	Fine scale	Body parts	1 m	1 μ s	Equalizer taps
7	Internal (radio)	HW, SW	.1 m	1 ns	Architecture

■ Table 4. The physical world inference hierarchy; HW, SW: hard/software.

eling. A UNIX stream can be a dynamic model, for example. Thus, cognitive radio could tell the network about its equalizer by binding its generic model to the dynamic model stream and reporting the results in KQML to the network. Since such values change as a function of time, RKRL will access (and log) signals as fluents [14] in order to detect regular patterns.

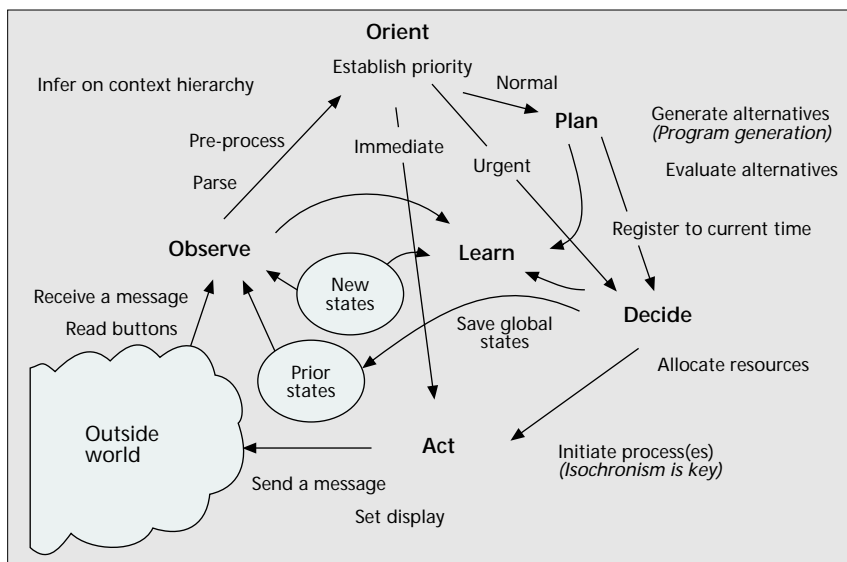
The current version of RKRL is implemented in Visual Basic attached to 41 Excel spreadsheets. Object linking and embedding from Excel allows RKRL to access almost any existing software as an executable model. Thus, instead of writing the large number of subordinate models needed for a comprehensive RKRL, the RKRL framework points to those that exist. In addition, KQML, SDL, IDL, and UML primitives are represented in RKRL. One of the benefits of this approach is an ability to express a given item in more than one standard way. Another benefit is the ability to parse expressions from other languages in order to extract existing knowledge for use in cognitive radio.

Spatial Inference Hierarchy

RKRL embeds a standard spatial inference hierarchy for space and time, as shown in Table 4. Each of the planes consists of objects with associated space-time properties. RKRL also declares ways in which the radio can autonomously obtain information about objects on that level. The global plane, for example, divides the Earth into large regions. The properties of the global plane change in annual cycles (e.g., through annual holiday patterns of travel). RKRL statements

at this level define the components of standard annual cycles including seasons of the year, weather, and holidays. The radio can get information about its user's interaction with the global plane by examining the user's travel itinerary. Other planes contain objects appropriate to that level of abstraction, including space-time characteristics and information sources.

The lowest level of this hierarchy represents the physical architecture of the software radio. It describes antennas, digital signal processors, memory (RAM and ROM), user interface devices, and so on in terms of physical capabilities and interconnections. Although there is nothing to preclude RKRL from invoking a complete HDL description of the radio, the goals of cognitive radio concern inference about higher-level aspects of radio etiquettes. RKRL micro-worlds in the internal plane therefore embed the architecture framework, applications programmer interface (API), and IDL of the SDR Forum.



■ Figure 6. The cognition cycle.

Model Matching

Detailed models of radio functions are embedded in RKRL 0.1 for each micro-world in which competence has been developed. As an example, Table 5 shows an executable model of the segmentation of a message into packets (from the segmentation micro-world).

If a cognitive radio sets the packet number to 0, the payload becomes SAMPL, the first five octets of the outgoing message. To determine the impact of changing the protocol, the system first copies the values of the model to temporary RKRL frames. It then changes the parameters to correspond to its hypothesized protocol (e.g., to a 7-byte payload). Finally, it compares the values of the model to the previously stored values to determine how the change of protocol will change the payload. It can thus independently reason about the correctness of a downloaded packetization module.

The Cognition Cycle

RKRL supports the cognition cycle illustrated in Fig. 6. The outside world provides stimuli. Cognitive radio parses these stimuli to extract the available contextual cues necessary for the performance of its assigned tasks. It might analyze GPS coordinates plus light and temperature to determine whether it is inside or outside a building. This type of processing occurs in the observe stage of the cognition cycle. Incoming and outgoing messages are parsed for content, including the content supplied to/by the user. This yields contextual cues necessary to infer the urgency of the communications and related internal tasks. This task is akin to topic spotting in natural language processing. Even relatively high word error rates can result in high probability of detection and low false alarm rate in detecting ordinary events. Thus, the radio "knows" it is going for a taxi ride (with some probability) if the user packets at the wireless information kiosk order a taxi. If the main battery has just been removed, however, the orient stage *immediately* acts to save data necessary for a graceful startup and to shut the system down. Loss of carrier on all available links (e.g., due to entering a building) can result in *urgent* steps to restore connectivity, such as scanning for an in-building PCS or RF LAN. Most other *normal* events might not require such time-sensitive responses, resulting in the plan-decide-act cycle. The act step consists of allocating computational and radio resources to subordinate (conventional radio) software and initiating tasks for specified amounts of time. RKRL also includes some forms of supervised and unsupervised learning.

Conclusion

Software radios provide a vast untapped potential to personalize services, but the contemporary process of modifying radio etiquettes is extremely labor-intensive. In part this is because there is no generally accepted way of representing radio knowledge. This limits the flexibility and responsiveness of the radio to the network and user. RKRL may provide some insights into how to better automate this process. Cognitive radio, built on RKRL, is envisioned as a competence system over the domain of radio resources and protocols. Its agent knowledge and inference mechanisms are under development, as is the initial critical mass definition of RKRL.

Finally, RKRL is designed to be used by software agents that have such a high level of competence, driven in part by a large store of a priori knowledge, that they may accurately be called

Parameter	Value
Message	SAMPLE MESSAGE
Payload size	40
Payload bytes	5
Packet number	1
Payload	E MES

■ Table 5. An Excel model of a message being parsed into packet payloads.

cognitive. This goal may be very far off, or may emerge from the current research program. Cognitive radio approaches the software radio as a micro-world. But radio engineering is such a large, complex world that it will require much effort to describe it in computationally accessible, useful ways. The present research is therefore offered as a mere baby step in a potentially interesting research direction.

References

- [1] J. Mitola, *Software Radio: Wireless Architecture for the 21st Century*, 103255.2507@compuserve.com, Mitola's STATISfaction, ISBN 0-9671233-0-5.
- [2] S. Kaufman, *At Home In The Universe*, New York: Oxford Univ. Press, 1995.
- [3] J. Mitola, "Cognitive Radio," Licentiate proposal, KTH, Stockholm, Sweden, Dec. 1998.
- [4] Location-Aware Computing and Communications Home Pages, KTH, Sweden, and Wollongong University, Australia: www.elec.uow.edu.au/people/staff/beadle/badge/location_aware.html
- [5] M. Henning, "Location Aware Mobile Applications Based on Directory Services," Mobicom '97, 1997.
- [6] ITU Z.100 Recs., "Specification and Description Language," Geneva, Switzerland, 1991.
- [7] Eriksson and Penker, *UML Toolkit*, New York: Wiley, 1998.
- [8] Mowbray and Zehavi, *The Essential CORBA*, Object Management Group, New York: Wiley, 1995.
- [9] Verilog, Object GEODE, Paris, France, 1998.
- [10] T. Finin, "KQML - A Language and Protocol for Knowledge and Information Exchange," Int'l. Conf. Building and Sharing of Very Large-Scale Knowledge Bases, Tokyo, Japan, Dec. 1993.
- [11] Genesereth and Fikes, "Knowledge Interchange Format Version 3.0 Reference Manual," Logic Group Rep. Logic 92-1, Stanford Univ., Palo Alto, CA, 1992.
- [12] SDR Forum Web Site, www.sdrforum.org
- [13] E. Davis, "The Naive Physics Perplex," *AI Mag.*, Menlo Park, CA, Winter 1998.
- [14] Cohen *et al.*, "Neo: Learning Conceptual Knowledge by Sensorimotor Interaction with an Environment," *Proc. Autonomous Agents '97*, 1997, www.acm.org

Biographies

JOSEPH MITOLA III (jmitola@jpo.org) is a consulting scientist with the MITRE Corporation. He has been a principal architect of the migration of billions of dollars of legacy telecommunications and information processing systems to commercially based, software-programmable, and highly interoperable open architecture systems of the future. At present, he is participating in an executive exchange program with the United States Department of Defense, where he is a program manager for the Defense Advance Research Projects Agency (DARPA). He is internationally recognized as the "Godfather" of the software radio, having coined the term in 1991 to signal the shift in radio technology from digital radio with "80 percent hardware and 20 percent software" to radios in which 80 percent of the value added of the radio is in the software. In addition to having published the first paper on software radios in 1992, he writes and teaches extensively on this technology. Prior to joining the MITRE Corporation in 1993, he was the chief scientist of Electronic System, E-Systems Melpar Division (now Raytheon, Falls Church), where he was responsible for research and technology development for advanced communications and electronic warfare systems for the United States Department of Defense. He previously held positions of technical responsibility with Harris Corporation, Advanced Decision Systems, and ITT Corporation. He began his career as an engineering student assistant with the United States Department of Defense in 1967. He holds a B.Sc. degree in electrical engineering from Northeastern University (highest honors) and a Master's in degree engineering from Johns Hopkins University. He is presently a Ph.D. candidate at KTH, the Royal Technical Institute, Stockholm, Sweden.

GERALD Q. MAGUIRE (maguire@it.kth.se) is head of the Computer Communications Systems Laboratory of the Royal Technical Institute (KTH), Stockholm, Sweden, where his research interests include environmentally aware computing (location-aware plus computer awareness of temperature, doors, heating and air conditioning, etc.) and fourth-generation wireless services and communications systems. During 1991 and 1992, he was program director for experimental systems with the U.S. National Science Foundation, Directorate for Computer and Information Science and Engineering, Division of Microelectronic Information Processing systems. From 1984 to 1993 he was an assistant professor at Columbia University. He received a Ph. D., computer science, from the University of Utah in 1983; an M.S. in computer science, University of Utah, 1981; and a B.A., magna cum laude, physics, Indiana University of Pennsylvania, 1975.