UNIT-IV

COGNITIVE RADIO ARCHITECTURE

Syllabus:


Introduction

Architecture is a comprehensive, consistent set of design rules by which a specified set of components achieves a specified set of functions in products and services that evolve through multiple design points over time.

This section introduces the fundamental design rules by which software-defined radio (SDR), sensors, perception, and automated machine learning (AML) may be integrated to create Aware, Adaptive, and Cognitive Radios (AACRs).

These SDRs will have better quality of information (QoI) through capabilities to Observe (sense, perceive), Orient, Plan, Decide, Act, and Learn (the so-called OOPDAL loop) in radio frequency (RF) and in the user domains.

By performing this integration, we will transition from merely adaptive to a demonstrably cognitive radio (CR).

Functions, Components, and Design Rules

The functions of AACR exceed those of SDR.

Reformulating the AACR <Self/> [the <Self/>, a self-referential subsystem that strictly embodies finite computing (e.g., no While or Until loops)] as a peer of its own <User/> establishes the need for added functions by which the <Self/> accurately perceives the local scene including the <User/> and autonomously learns to tailor the information services to the specific <User/> in the current RF and physical <Scene/>.

AACR Functional Component Architecture

The SDR components and the related cognitive components of iCR(ideal CR) appear in figure below. The cognition components describe the SDR in RXML so that the <Self/> can know that it is a radio and that its goal is to achieve high QoI tailored to its own users.

The CRA is expressed in Radio eXtensible Markup Language (RXML). RXML intelligence includes a priori radio background and user stereotypes as well as knowledge of RF and space–time <Scenes/> perceived and experienced.
This includes both structured reasoning with iCR peers and cognitive wireless networks (CWNs), and ad hoc reasoning with users, all the while learning from experience.

![SDR Components](image)

**The CRA augments SDR with computational intelligence and learning capacity**

*Fig 1*

**SDR Components**

SDRs include a hardware platform with RF access and computational resources, plus at least one software-defined personality.

The SDR Forum has defined its SCA (Software Communications Architecture) and the Object Management Group (OMG) has defined its SRA (Software Radio Architecture).

These are similar fine-grained architecture constructs enabling reduced-cost wireless connectivity with next-generation plug-and-play.

These SDR architectures are defined in Unified Modeling Language (UML) object models, Common Object Request Broker Architecture (CORBA), Interface Design Language (IDL), and XML descriptions of the UML models.
**AACR Node Functional Components**

A simple CRA includes the functional components shown in figure below

![Minimal AACR node architecture](image)

**Fig 2**

A functional component is a black box to which functions have been allocated, but for which implementation is not specified.

These functional components are as follows:

1. The *user sensory perception* (SP), which includes haptic, acoustic, and video sensing and perception functions.
2. The local *environment* sensors (location, temperature, accelerometer, compass, etc.).
3. The *system applications* (sys apps) media-independent services such as playing a network game.
4. The *SDR* functions which include RF sensing and SDR applications.
5. The *cognition* functions (symbol grounding for system control, planning, and learning).
6. The *local effector* functions (speech synthesis, text, graphics, and multimedia displays).

These functional components are embodied on an iCR platform, a hardware realization of the six functions.

AACR consists of six functional components: user SP, environment, effectors, SDR, sys apps, and cognition. Those components of the <Self/> enable external communications and internal reasoning about the <Self/> by using the RXML syntax.

In part, this equation

```
<CR-Platform>
 <Functional-Components>
 <User SP/> <Environment/> <Effectors/> <SDR/> <Sys Apps/> <Cognition/>
 </Functional-Components>
 <Self>
</CR-Platform>
```
states that the hardware–software platform and the functional components of the AACR are independent. Platform-independent computer languages such as Java are well understood.

This ontological perspective envisions platform independence as an architecture design principle for AACR.

*Ontology* is the philosophical study of the nature of being, becoming, existence or reality as well as the basic categories of being and their relations.

**Design Rules Include Functional Component Interfaces**

The six functional components (see Tables 1(a) and 1(b)) imply associated functional interfaces.

In architecture, design rules may include a list of the quantities and types of components as well as the interfaces among those components. This section addresses the interfaces among the functional components.

The AACR N-squared diagram of Table 1(a) characterizes AACR interfaces.

These constitute an initial set of AACR APIs. In some ways, these APIs augment the established SDR APIs.

This is entirely new and much needed in order for basic AACRs to accommodate even the basic ideas of the Defense Advanced Research Projects Agency (DARPA) NeXt-Generation (XG) radio communications program.

<table>
<thead>
<tr>
<th>From\to</th>
<th>User SP</th>
<th>Environment</th>
<th>Sys apps</th>
<th>SDR</th>
<th>Cognition</th>
<th>Effectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>User SP</td>
<td>1</td>
<td>7</td>
<td>13 PAa</td>
<td>19</td>
<td>25 PAb</td>
<td>31</td>
</tr>
<tr>
<td>Environment</td>
<td>2</td>
<td>8</td>
<td>14 SAa</td>
<td>20</td>
<td>26 PAb</td>
<td>32</td>
</tr>
<tr>
<td>Sys apps</td>
<td>3</td>
<td>9</td>
<td>15 SCMa</td>
<td>21 SDa</td>
<td>27 PDCa,b</td>
<td>33 PEMa</td>
</tr>
<tr>
<td>SDR</td>
<td>4</td>
<td>10</td>
<td>16 PDA</td>
<td>22 SD</td>
<td>28 PCb</td>
<td>34 SD</td>
</tr>
<tr>
<td>Cognition</td>
<td>5 PEBb</td>
<td>11 PEBb</td>
<td>17 PCA,b</td>
<td>23 PEBb</td>
<td>29 SCb</td>
<td>35 PEBb</td>
</tr>
<tr>
<td>Effectors</td>
<td>6 SC</td>
<td>12</td>
<td>18a</td>
<td>24</td>
<td>30 PCDb</td>
<td>36</td>
</tr>
</tbody>
</table>

P: primary; A: afferent; E: efferent; C: control; M: multimedia; D: data; S: secondary; others not designated P or S are ancillary.

aInformation services API; bCAPI.

**Table 1(a)**

In other ways, these APIs supersede the existing SDR APIs. In particular, the SDR user interface becomes the user sensory and effector API.

User sensory APIs include acoustics, voice, and video, and the effector APIs include speech synthesis to give the AACR <Self/> its own voice.

These interface changes enable the AACR to sense the situation represented in the environment, to interact with the user, and to access radio networks on behalf of the user in a situation-aware way.
<table>
<thead>
<tr>
<th>Note number</th>
<th>Process interface</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User SP–User SP</td>
<td>Cross-media correlation interfaces (video-acoustic, haptic-speech, etc.) to limit search and reduce uncertainty (e.g., if video indicates user is not talking, acoustics may be ignored or processed less aggressively for command inputs than if user is speaking).</td>
</tr>
<tr>
<td>2</td>
<td>Environment–User SP</td>
<td>Environment sensors parameterize user sensor-perception. Temperature below freezing may limit video.</td>
</tr>
<tr>
<td>3</td>
<td>Sys apps–User SP</td>
<td>Sys apps may focus scene perception by identifying entities, range, expected sounds for video, audio, and spatial perception processing.</td>
</tr>
<tr>
<td>4</td>
<td>SDR–User SP</td>
<td>SDR applications may provide expectations of user input to the perception system to improve probability of detection and correct classification of perceived inputs.</td>
</tr>
<tr>
<td>5</td>
<td>Cognition–User SP</td>
<td>This is the primary control efferent path from cognition to the control of the user SP subsystem, controlling speech recognition, acoustic signal processing, video processing, and related SP. Plans from cognition may set expectations for user scene perception, improving perception.</td>
</tr>
<tr>
<td>6</td>
<td>Effectors–User SP</td>
<td>Effectors may supply a replica of the effect to user perception so that self-generated effects (e.g., synthesized speech) may be accurately attributed to the &lt;Self/&gt; conditons, validated as having been expressed, and/or canceled from the scene perception to limit search.</td>
</tr>
<tr>
<td>7</td>
<td>User SP–Environment</td>
<td>Perception of rain, buildings, indoor/outdoor can set GPS integration parameters.</td>
</tr>
<tr>
<td>8</td>
<td>Environment–Environment</td>
<td>Environment sensors would consist of location sensing such as GPS or GLONASS: ambient temperature; light level to detect inside versus outside locations; possibly smell sensors to detect spoiled food, fire, etc. There seems to be little benefit in enabling interfaces among these elements directly.</td>
</tr>
<tr>
<td>9</td>
<td>Sys apps–Environment</td>
<td>Data from the sys apps to environment sensors would also be minimal.</td>
</tr>
<tr>
<td>10</td>
<td>SDR–Environment</td>
<td>Data from the SDR personalities to the environment sensors would be minimal.</td>
</tr>
<tr>
<td>11</td>
<td>Cognition–Environment (primary control path)</td>
<td>Data from the cognition system to the environment sensors controls those sensors, turning them on and off, setting control parameters, and establishing internal paths from the environment sensors.</td>
</tr>
<tr>
<td>12</td>
<td>Effectors–Environment</td>
<td>Data from effectors directly to environment sensors would be minimal.</td>
</tr>
<tr>
<td>13</td>
<td>UserSP–Sys apps</td>
<td>Data from the user SP system to sys apps is a primary efferent path for multimedia streams and entity states that effect information services implemented as sys apps. Speech, images, and video to be transmitted move along this path for delivery by the relevant sys apps or information service to the relevant wired or SDR communications path. Sys apps overcomes the limitations of individual paths by maintaining continuity of conversations, data integrity, and application coherence (e.g., for multimedia games). Whereas the cognition function sets up, tears down, and orchestrates the sys apps, the primary API between the user scene and the information service consists of this interface and its companions—the environment afferent path, the effector efferent path, and the SDR afferent and efferent paths.</td>
</tr>
<tr>
<td>14</td>
<td>Environment–Sys apps</td>
<td>Data on this path assists sys apps in providing location awareness to services.</td>
</tr>
</tbody>
</table>

(Continued)
<table>
<thead>
<tr>
<th>Note number</th>
<th>Process interface</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Sys apps–Sys apps</td>
<td>Different information services interoperate by passing control information through the cognition interfaces and by passing domain multimedia flows through this interface. The cognition system sets up and tears down these interfaces.</td>
</tr>
<tr>
<td>16</td>
<td>SDR–Sys apps</td>
<td>This is the primary afferent path from external communications to the AACR. It includes control and multimedia information flows for all the information services. Following the SDR Forum’s SCA, this path embraces wired as well as wireless interfaces.</td>
</tr>
<tr>
<td>17</td>
<td>Cognition–Sys apps</td>
<td>Through this path, the AACR &lt;Self&gt; exerts control over the information services provided to the &lt;User/&gt;.</td>
</tr>
<tr>
<td>18</td>
<td>Effectors–Sys apps</td>
<td>Effectors may provide incidental feedback to information services through this afferent path, but the use of this path is deprecated. Information services are supposed to control and obtain feedback through the mediation of the cognition subsystem.</td>
</tr>
<tr>
<td>19</td>
<td>User SP–SDR</td>
<td>Although the SP system may send data directly to the SDR subsystem (e.g., to satisfy security rules that user biometrics be provided directly to the wireless security subsystem), the use of this path is deprecated. Perception subsystem information is supposed to be interpreted by the cognition system so that accurate information, not raw data, can be conveyed to other subsystems.</td>
</tr>
<tr>
<td>20</td>
<td>Environment–SDR</td>
<td>Environment sensors such as GPS historically have accessed SDR waveforms directly (e.g., providing timing data for air interface signal generation). The cognition system may establish such paths in cases where cognition provides little or no value added, such as providing a precise timing reference from GPS to an SDR waveform. The use of this path is deprecated because all of the environment sensors, including GPS, are unreliable. Cognition has the capability to “de-glitch” GPS (e.g., recognize from video that the &lt;Self&gt; is in an urban canyon and therefore not allow GPS to report directly, but report to the GPS subscribers, on behalf of GPS, location estimates based perhaps on landmark correlation, dead reckoning, etc.).</td>
</tr>
<tr>
<td>21</td>
<td>Sys apps–SDR</td>
<td>This is the primary efferent path from information services to SDR through the services API.</td>
</tr>
<tr>
<td>22</td>
<td>SDR–SDR</td>
<td>The linking of different wireless services directly to each other is deprecated. If an incoming voice service needs to be connected to an outgoing voice service, there should be a bridging service in sys apps through which the SDR waveforms communicate with each other. That service should be set up and taken down by the cognition system.</td>
</tr>
<tr>
<td>23</td>
<td>Cognition–SDR</td>
<td>This is the primary control interface, replacing the control interface of the SDR SCA and the OMG SRA.</td>
</tr>
<tr>
<td>24</td>
<td>Effectors–SDR</td>
<td>Effectors such as speech synthesis and displays should not need to provide state information directly to SDR waveforms, but if needed, the cognition function should set up and tear down these interfaces.</td>
</tr>
<tr>
<td>25</td>
<td>User SP–Cognition</td>
<td>This is the primary afferent flow for the results from acoustics, speech, images, video, video flow, and other sensor-perception subsystems. The primary results passed across this interface should be the specific states of &lt;Entities/&gt; in the scene, which would include scene characteristics such as the recognition of landmarks, known vehicles, furniture, and the like. In other words, this is the interface by which the presence of &lt;Entities/&gt; in the local scene is established and their characteristics are made known to the cognition system.</td>
</tr>
<tr>
<td>26</td>
<td>Environment–Cognition</td>
<td>This is the primary afferent flow for environment sensors.</td>
</tr>
</tbody>
</table>

(Continued)
Near-Term Implementations

One way to implement this set of functions is to embed into an SDR a reasoning engine, such as a rule base with an associated inference engine, as the cognition function.

If the effector functions control parts of the radio, then we have the simplest AACR based on the simple six-component architecture of Fig 1.
Such an approach may be sufficient to expand the control paradigm from today’s state machines with limited flexibility to tomorrow’s AACR control based on reasoning over more complex RF states and user situations.

Such simple approaches may well be the next practical steps in AACR evolution from SDR toward iCR.

**The Cognition Components**

Fig 1 shows three computational intelligence aspects of CR:

1. Radio knowledge—RXML:RF
2. User knowledge—RXML:User
3. The capacity to learn

The minimalist architecture of Fig 2 and the functional interfaces of Tables 1(a) and 1(b) do not assist the radio engineer in structuring knowledge, nor do they assist much in integrating Machine Learning (ML) into the system.

**Radio Knowledge in the Architecture**

Radio knowledge has to be translated from the classroom and engineering teams into a body of computationally accessible, structured technical knowledge about radio.

The SCA structures the technical knowledge of the radio components into UML and XML.

RXML will enable the structuring of sufficient RF and user world knowledge to build advanced wireless-enabled or enhanced information services.

Thus, whereas the SRA and SCA focus on building radios, RXML focuses on using radios.

The World Wide Web (WWW) is now sprouting with computational ontologies some of which are nontechnical but include radio, such as the open Cyc ontology.

[Cyc is an artificial intelligence (AI) project that attempts to assemble an encyclopedic comprehensive ontology and database of everyday common sense knowledge, with the goal of enabling AI applications to perform human-like reasoning]

They bring the radio domain into the Semantic Web, which helps people know about radio.

This informal knowledge lacks the technical scope, precision, and accuracy of authoritative radio references such as the European Telecommunications Standards Institute (ETSI) documents defining the Global System for Mobile Communications (GSM) and the International Telecommunication Union (ITU) definitions of, for example 3GPP.

The capabilities required for an AACR node to be a cognitive entity are to sense, perceive, orient, plan, decide, act, and learn.

Table below is illustrative and not comprehensive, but it characterizes the technical issues that drive an information-oriented AACR node architecture.
User Knowledge in the Architecture

User knowledge is formalized at the level of abstraction and degree of detail necessary to give the CR the ability to acquire, from its owner and other designated users, the user knowledge relevant to information services incrementally.

Incremental knowledge acquisition was motivated in the introduction to ML by describing how a frequent occurrence with similar activity sequences identifies learning opportunities.

ML machines may recognize these opportunities for learning through joint probability statistics <Histogram/>.

Effective use cases clearly identify the classes of user and the specific knowledge learned to customize envisioned services.

Use cases may also supply sufficient initial knowledge to render incremental ML not only effective, but also—if possible—enjoyable to the user.

This knowledge is defined in RXML:User. As with RF knowledge, the capabilities required for an AACR node to be a cognitive entity are to OOPDAL. To relate a use case to these capabilities, one extracts
specific and easily recognizable <Anchors/> for stereotypical situations observable in diverse times, places, and situations. One expresses the anchor knowledge in RXML for use on the AACR node.

**Cross-domain Grounding for Flexible Information Services**
The knowledge about radio and about user needs for wireless services must be expressed internally in a consistent form so that information services relationships may be autonomously discovered and maintained by the <Self/> on behalf of the <User/>. Figure below shows relationships among user and RF domains.

Staying better connected requires the normalization of knowledge between <User/> and <RF/> domains.

If, for example, the <User/> says, “What’s on one oh seven, seven,” near the Washington, DC, area, then the dynamic <User/> ontology should enable the CR to infer that the user is talking about the current frequency modulation (FM) radio broadcast, the units are in MHz, and the user wants to know what is on WTOP.

If it can’t infer this, then it should ask the user or discover by first dialing a reasonable default, such as 107.7 FM, a broadcast radio station, and asking, “Is this the radio station you want?”

Steps 4, 5, and 6 in above fig all benefit from agreement across domains on how to refer to radio services.

Optimizing behavior to best support the user requires continually adapting the <User/> ontology with repeated regrounding of terms in the <User/> domain to conceptual primitives and actions in the <RF/> domain.
Specific methods of cross-domain grounding with associated architectural features include:

1. `<RF/> to <User/> shaping dialog to express precise `<RF/>` concepts to non-expert users in an intuitive way, such as:
   (a) Grounding: “If you move the speaker box a little bit, it can make a big difference in how well the remote speaker is connected to the wireless transmitter on the television (TV).”
   (b) AACR information architecture: Include facility for rich set of synonyms to mediate cognition–NL–synthesis interface (`<Antenna> ≡ <Wireless-remote-speaker> ≡ “Speaker box”).

2. `<RF/> to <User/> learning jargon to express `<RF/>` connectivity opportunities in `<User/>` terms:
   (a) Grounding: “Tee oh pee” for “WTOP,” “Hot ninety-two” for “FM 92.3.”
   (b) AACR information architecture: NL–visual facility for single-instance update of user jargon.

3. `<User/> to `<RF/> relating values to actions: Relate `<User/>` expression of values (“low-cost”) to features of situations (“normal”) that are computable (`<NOT> (<CONTAINS> <Situation> <Unusual/>)<..>`) and that relate directly to `<RF/>` domain decisions:
   (a) Grounding: Normally wait for free wireless local area network (WLAN) for big attachment; if situation is `<Unusual/>`, ask if user wants to pay for 3G.
   (b) AACR information architecture: Associative inference hierarchy that relates observable features of a `<Scene/>` to user sensitivities, such as `<Late-work/>` => `<Unusual/>`; “The President of the company needs this” => `<Unusual/>` because “President” => `<VIP/>` and `<VIP/>` is not in most scenes.

**Self-referential Components**

The cognition component must to assess, manage, and control all of its own resources, including validating downloads. Thus, in addition to `<RF/>` and `<User/>` domains, RXML must describe the `<Self/>`, defining the AACR architecture to the AACR itself in RXML.

**Self-referential Inconsistency**

This class of self-referential reasoning is well known in the theory of computing to be a potential black hole for computational resources.

Specifically, any Turing capable (TC) computational entity that reasons about itself can encounter unexpected Gödel-Turing situations from which it cannot recover.
Thus, TC systems are known to be “partial”—only partially defined because the result obtained when attempting to execute certain classes of procedure is not definable (the computing procedure will never terminate).

To avoid this paradox, CR architecture mandates the use of only “total” functions, typically restricted to bounded minimalization. Watchdog “step-counting” functions or timers must be in place in all its self-referential reasoning and radio functions.

**Watchdog Timer**

A **watchdog timer** (WDT) is a hardware timer that automatically generates a system reset if the main program neglects to periodically service it. It is often used to automatically reset an embedded device that hangs because of a software or hardware fault.

Without the reliable watchdog timer in the architecture and without this proof to establish the rules for acceptable computing constructs on CRs, engineers and computer programmers would build CRs that would crash in extremely unpredictable ways as their adaptation algorithms got trapped in unpredictable unbounded self-referential loops.

This timer-based finite computing regime also works for user interfaces because users will not wait forever before changing the situation (e.g., by shutting off the radio or hitting another key); and the CR can always kind of throw up its hands and ask the user to take over.

Thus, with a proof of stability based on the theory of computing, the CRA structures systems that not only can modify themselves, but also can do it in such a way that they are not likely to induce nonrecoverable crashes from the “partial” property of self-referential computing.

**Flexible Functions of the Component Architecture**

These functions of the architecture shown in Table below are not different from those of the six-component architecture, but represent varying degrees of instantiation of the six components.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Function</th>
<th>Examples (RF; vision; speech; location; motion)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cognition</td>
<td>Monitor and learn</td>
<td>Get to know user’s daily patterns and model the local RF scene over space, time, and situations</td>
</tr>
<tr>
<td>Adaptation</td>
<td>Respond to changing environment</td>
<td>Use unused RF, protect owner’s data</td>
</tr>
<tr>
<td>Awareness</td>
<td>Extract information from sensor domain</td>
<td>Sense or perceive</td>
</tr>
<tr>
<td>Perception</td>
<td>Continuously identify knowns, unknowns, and backgrounds in the sensor domain</td>
<td>TV channel; depth of visual scene, identity of objects; location of user, movement and speed of &lt;Self/&gt;</td>
</tr>
<tr>
<td>Sensing</td>
<td>Continuously sense and preprocess single-sensor field in single-sensory domain</td>
<td>RF FFT; binary vision; binaural acoustics; GPS; accelerometer; etc.</td>
</tr>
</tbody>
</table>
Consider the following degrees of architecture instantiations:

*Cognition functions of radio* entail the monitoring and structuring knowledge of the behavior patterns of the *<Self/>*, the *<User>*, and the environment (physical, user situation, and radio) to provide information services, learning from experience to tailor services to user preferences and differing radio environments.

- *Adaptation functions of radio* respond to a changing environment, but can be achieved without learning if the adaptation is preprogrammed.

- *Awareness functions of radio* extract usable information from a sensor domain. Awareness stops short of perception. Awareness is required for adaptation, but awareness does not guarantee adaptation. For example, embedding a GPS receiver into a cell phone makes the phone more location aware, but unless the value of the current location is actually used by the phone to do something that is location dependent, the phone is not location adaptive, only location aware. These functions are a subset of the CRA that enable adaptation.

- *Perception functions of radio* continuously identify and track knowns, unknowns, and backgrounds in a given sensor domain. Backgrounds are subsets of a sensory domain that share common features that entail no particular relevance to the functions of the radio. For a CR that learns initially to be a single-owner radio, in a crowd, the owner is the object that the radio continuously tracks in order to interact when needed. Worn from a belt as a cognitive wireless PDA (CWPDA), the iCR perception functions may track the entities in the scene. The nonowner entities comprise mostly irrelevant background because no matter what interactions may be offered by these entities, the CR will not obey them—only the interactions of the perceived owner. These functions are a subset of the CRA that enable cognition.

- *The sensory functions of radio* entail those hardware and/or software capabilities that enable a radio to measure features of a sensory domain. Sensory domains include anything that can be sensed, such as audio, video, vibration, temperature, time, power, fuel level, ambient light level, sun angle (e.g., through polarization), barometric pressure, smell, and anything else imaginable. Sensory domains for vehicular radios may be much richer, if less personal, than those of wearable radios. Sensory domains for fixed infrastructure could include weather features such as ultraviolet sunlight, wind direction and speed, humidity, traffic flow rate, or rain rate. These functions are a subset of the CRA that enable perception.

**Design Rules**

The following design rules circumscribe the way cognitive radio functions are mapped to the components of a wireless PDA within the envisioned architecture, as follows:

1. The cognition functions shall maintain an explicit topological model of the world, of the environment (including the user, the physical environment, and the radio networks, and of the internal states of the radio). Context shall be axiomatically represented using the topological model.

2. The model of the world shall follow an explicit axiomatic treatment of time, space, radio frequency, radio propagation, and the identity of entities.

3. Models shall be represented in an open architecture radio knowledge representation language suited to the representation of radio knowledge (e.g. RKRL 0.3). That language shall support topological properties and axiomatic models of cognitive radio.
4. The cognition functions shall maintain location awareness, including the sensing of location from
global positioning satellites, and sensing position from local wireless sensors and networks.
Location shall be an element of context.

5. The cognition functions shall maintain awareness of time to the accuracy necessary to support the
control of radio functions. Time shall be an element of context.

6. The cognition functions shall maintain an awareness of the identity of the PDA, of its primary
user, and of other legitimate users designated by the primary user. Current user shall be an
element of context.

7. The cognition functions shall reliably infer the user's communications context and apply that
knowledge to the provisioning of wireless access by the SDR function.

8. The cognition functions shall model the propagation of its own radio signals with sufficient
fidelity to estimate interference to other spectrum users. The cognition function shall also assure
that interference is within limits specified by the spectrum use protocols in effect in its location
(e.g. in spectrum rental protocols). It shall defer to the wireless network in contexts where the
network manages interference.

9. The cognition functions shall model the domain of applications running on the host platform,
sufficient to infer the parameters needed to support the application. Parameters modeled include
QoS, data rate, probability of link closure (Grade of Service), and space x time x context domain
within which wireless support is needed.

10. The cognition functions shall configure and manage the SDR assets to include hardware
resources, software personalities, and functional capabilities as a function of network constraints
and use context.

11. The cognition functions shall administer the computational resources of the platform. The
management of software radio resources may be delegated to an appropriate SDR function (e.g.
the SDR Forum domain manager). Constraints and parameters of those SDR assets shall be
modeled by the cognition functions. The cognition functions shall assure that the computational
resources allocated to applications, interfaces, cognition and SDR functions are consistent with
the user communications context.

12. The cognition functions shall represent degree of belief in external stimuli and in inferences. A
certainty calculus shall be employed consistently in reasoning about uncertain information.

13. The cognition functions shall recognize preemptive actions taken by the network and/or the user.
In case of conflict, the cognition functions shall defer the control of applications, interfaces,
and/or SDR assets to the network or to the primary user, according to an appropriate operations
assurance protocol.
The Cognition Cycle

The CRA comprises a set of design rules by which the cognitive level of information services may be achieved by a specified set of components in a way that supports the cost-effective evolution of increasingly capable implementations over time. The cognition subsystem of the architecture includes an inference hierarchy and the temporal organization and flow of inferences and control states—the cognition cycle.

The cognition cycle developed for CR1 is illustrated in Figure 14.5.

![Diagram: Simplified cognition cycle. The observe, orient, plan, decide, act (OOPDA) loop is a primary cycle; learning, planning, and sensing the outside world are crucial phases of the larger OOPDA-loop (© Dr. Joseph Mitola III, used with permission).]

This cycle implements the capabilities required of iCR in a reactive sequence.

Stimuli enter the CR as sensory interrupts, dispatched to the cognition cycle for a response.

Such an iCR continually observes (senses and perceives) the environment, orients itself, creates plans, decides, and then acts.

In a single-processor inference system, the CR’s flow of control may also move in the cycle from observation to action.

In a multiprocessor system, temporal structures of sensing, preprocessing, reasoning, and acting may be parallel and complex. Special features synchronize the inferences of each phase.

The tutorial code all works on a single processor in a rigid inference sequence defined in Figure 14.5.
This process is called the “wake epoch” because the primary reasoning activities during this large epoch of time are reactive to the environment.

We will refer to “sleep epochs” for power-down conditions, “dream epochs” for performing computationally intensive pattern recognition and learning, and “prayer epochs” for interacting with a higher authority such as network infrastructure.

During the wake epoch, the receipt of a new stimulus on any of a CR’s sensors or the completion of a prior cognition cycle initiates a new primary cognition cycle.

The CR observes its environment by parsing incoming information streams. These can include monitoring and speech-to-text conversion of radio broadcasts (e.g., the Weather Channel, stock ticker tapes, etc.).

Any RF-LAN or other short range wireless broadcasts that provide services awareness information may be also parsed.

In the observation phase, a CR also reads location, temperature, and light level sensors, among other parameters, to infer the user’s communications context.

**Observe (Sense and Perceive)**

The iCR senses and perceives the environment (via “observation phase” code) by accepting multiple stimuli in many dimensions simultaneously and by binding these stimuli—all together or more typically in subsets—to prior experience so that it can subsequently detect time-sensitive stimuli and ultimately generate plans for action.

Observe phase comprises both the user SP and the environment (RF and physical) sensor subsystems.

**Orient**

Orient phase is part of the cognition component.

The orient phase determines the significance of an observation, by binding the observation to a previously known set of stimuli of a “scene.”

The orient phase contains the internal data structures that constitute the equivalent of the short-term memory (STM) that people use to engage in a dialog without necessarily remembering everything with the same degree of long-term memory (LTM).

In the CRA, the transfer from STM to LTM is mediated by the sleep cycle in which the contents of STM since the last sleep cycle are analyzed both internally and with respect to existing LTM.

Matching of current stimuli to stored experience may be achieved by “stimulus recognition” or by “binding.” The orient phase is the first collection of activity in the cognition component.

**Stimulus Recognition**

Stimulus recognition occurs when there is an exact match between a current stimulus and a prior experience.
The CR1 prototype is continually recognizing exact matches and recording the number of exact matches that occurred along with the time measured in the number of cognition cycles between the last exact match.

By default, the response to a given stimulus is to merely repeat that stimulus to the next layer up the inference hierarchy for aggregation of the raw stimuli.

But if the system has been trained to respond to a location, a word, an RF condition, a signal on the power bus, or some other parameter, it may either react immediately or plan a task in reaction to the detected stimulus. If that reaction were in error, then it may be trained to ignore the stimulus, given the larger context, which consists of all the stimuli and relevant internal states, including time.

**Note:** Sometimes, the orient phase causes an action to be initiated immediately as a “reactive” stimulus–response behavior. A power failure, for example, might directly invoke an act that saves the data.

**Binding**

Binding occurs when there is a nearly exact match between a current stimulus and a prior experience and very general criteria for applying the prior experience to the current situation are met.

Binding also determines the priority associated with the stimuli.

Better binding yields higher priority for autonomous learning, whereas less-effective binding yields lower priority for the incipient plan.

**Plan**

An incoming network message would normally be dealt with by generating a plan (in the plan phase, the “normal” path). Such planning includes plan generation. The plan phase should also include reasoning about time.

Typically, reactive responses are preprogrammed or defined by a network (i.e., the CR is “told” what to do), whereas other behaviors might be planned.

A stimulus may be associated with a simple plan as a function of planning parameters with a simple planning system.

Open source planning tools enable the embedding of planning subsystems into the CRA, enhancing the plan component.

**Decide**

The decide phase selects among the candidate plans. The radio might have the choice to alert the user to an incoming message (e.g., behave like a pager) or to defer the interruption until later (e.g., behave like a secretary who is screening calls during an important meeting).
**Act**

Acting initiates the selected processes using effector modules. Effectors may access the external world or the CR’s internal states.

**Externally Oriented Actions**

Access to the external world consists primarily of composing messages to be spoken into the local environment or expressed in text form locally or to another CR or CN using the Knowledge Query and Manipulation Language (KQML), Radio Knowledge Representation Language (RKRL), Web Ontology Language (OWL), Radio eXtensible Markup Language (RXML), or some other appropriate knowledge interchange standard.

**Internally Oriented Actions**

Actions on internal states include controlling machine-controllable resources such as radio channels.

The CR can also affect the contents of existing internal models, such as adding a model of stimulus–experience–response (serModel) to an existing internal model structure.

The new concept itself may assert-related concepts into the scene.

The experience may be reactively integrated into RXML knowledge structures as well, provided the reactive response encodes them properly.

**Learning**

Learning is a function of perception, observations, decisions, and actions.

Initial learning is mediated by the observe phase perception hierarchy in which all SP are continuously matched against all prior stimuli to continually count occurrences and to remember time since the last occurrence of the stimuli from primitives to aggregates.

Learning also occurs through the introduction of new internal models in response to existing models and case-based reasoning (CBR) bindings.

In general, there are many opportunities to integrate ML into AACR. Each of the phases of the cognition cycle offers multiple opportunities for discovery processes, such as <Histogram/>, as well as many other ML approaches.

Finally, a learning mechanism occurs when a new type of serModel is created in response to an action to instantiate an internally generated serModel.

For example, prior and current internal states may be compared with expectations to learn about the effectiveness of a communications mode, instantiating a new mode specific serModel.
The Inference Hierarchy

The phases of inference from observation to action show the flow of inference, a top-down view of how cognition is implemented algorithmically.

The inference hierarchy is the part of the algorithm architecture that organizes the data structures.

An illustrative inference hierarchy includes layers from atomic stimuli at the bottom to information clusters that define action contexts, as shown in Figure 14.6

![Figure 14.6: Standard inference hierarchy](image)

The pattern of accumulating elements into sequences begins at the bottom of the hierarchy.

Atomic Stimuli

Atomic stimuli originate in the external environment and are sensed and preprocessed by the sensory subsystems, which include sensors of the RF environment (e.g., radio receiver and related data and information processing) and of the local physical environment, including acoustic, video, and location sensors.

Atomic symbols are the elementary stimuli extracted from the atomic stimuli.

Atomic symbols may result from a simple noise-riding threshold algorithm, such as the squelch circuit in RF that differentiates signal from noise. [Acoustic signals may be differentiated from simple background noise this way, but generally the result is the detection of a relatively large speech epoch that contains various kinds of speech energy. Thus, further signal processing is typically required in a preprocessing subsystem to isolate atomic symbols.]

The transformation from atomic stimuli to atomic symbols is the job of the sensory preprocessing system.

Primitive Sequences: Words and Dead Time

The accumulation of sequences of atomic symbols forms primitive sequence.

Primitive sequences have spatial and/or temporal coincidence, standing out against the background (or noise), but there may be no particular meaning in that pattern of coincidence.
The key question at this level of the data structure hierarchy is the sequence boundary.

The simplest situation is one in which a distinguished atomic symbol separates primitive sequences, which is exactly the case with white space between words in typed text.

A text based ML system may be white space to separate a text stream into primitive sequences.

**Basic Sequences**

Basic sequences are space–time–spectrum sequences that entail the communication of discrete messages.

The pattern of aggregation is repeated vertically at the levels corresponding to words, phrases, dialogs, and scenes.

The data structures generated by processing nodes create the concept hierarchy. These are the reinforced hierarchical sequences. They are reinforced by the inherent counting of the number of times each atomic or aggregated stimulus occurs.

The phrase level typically contains or implies a verb (the verb “to be” may be implied if no other verb is implicit).

**Sequences cluster**

These discrete messages (e.g., phrases) are typically defined with respect to an ontology of the primitive sequences (e.g., definitions of words). Sequences cluster together because of shared properties. For example, phrases that include words such as “hit,” “pitch,” “ball,” and “out” may be associated with a discussion of a baseball game. Knowledge Discovery in Databases (KDD) and the Semantic Web offer approaches for defining, or inferring, the presence of such clusters from primitive and basic sequences.

**Context cluster**

A scene is a context cluster, a multidimensional space–time–frequency association, such as a discussion of a baseball game in the living room on a Sunday afternoon. Such clusters may be inferred from unsupervised ML (e.g., using statistical methods or nonlinear approaches such as SVMs).

Although presented here in a bottom-up fashion, there is no reason to limit multidimensional inference to the top layers of the inference hierarchy. The lower levels of the inference hierarchy may include correlated multisensor data.

**For example,** a word may be characterized as a primitive acoustic sequence coupled to a primitive sequence of images of a person speaking that word. In fact, taking the cue that infants seem to thrive on multisensory stimulation, the key to reliable ML may be the use of multiple sensors with multisensor correlation at the lowest levels of abstraction.
Cognition functions are implemented via cognition elements consisting of data structures, processes, and flows, which may be modeled as topological maps over the abstract domains identified in Figure 14.12.

The \textless Self/\textgreater\ is an entity in the world, whereas the internal organization of the \textless Self/\textgreater (annotated PDA in the figure) is an abstraction that models the \textless Self/\textgreater.

The hierarchy of words, phrases, and dialogs from sensory data to scenes is not inconsistent with visual perception.

Words correspond to visual entities; phrases to detectable movement and juxtaposition of entities in a scene.

Dialogs correspond to a coherent sequence of movement within the scope of a scene, such as walking across the room.

Occlusion may be thought of as a dialog in which the room asserts itself in part of the scene while observable walking corresponds to assertion of the object.

The model data structures may be read as generalized words, phrases, dialogs, and scenes that may be acoustic, visual, or perceived in other sensory domains (e.g., infrared).

These structures refer to set-theoretic spaces consisting of a set $X$ and a family of subsets $Ox$ that contain $\{X\}$ and $\{\}$, the null set, and that are closed under union and countable intersection.
Data- and knowledge-storage spaces are shown as rectangles (e.g., dialog states, plans), whereas processing elements that transform sets are modeled as homeomorphisms, or topology-preserving maps, shown as directed graphs (e.g., Π) in this figure.

**CRA Topological Maps**

The processing elements of the architecture are modeled topological maps, as shown in Figure 14.12:

1. The input map $\vartheta$ consists of components that transform external stimuli to the internal data structure sensory data.
2. The transformation $\Theta$ consists of entity recognition (via acoustic, optical, and other sensors), lower-level software radio (SWR) waveform interface components, and so forth, that create streams of primitive-reinforced sequences. The model includes maps that form successively higher-level sequences from the data on the immediately lower level.
3. Reasoning components include the map $\rho$ that identifies the best match of known sequences to novel sequences. These are bound to scene variables by projection components, $\Pi$. The maps $\vartheta$, $\Theta$, $\rho$, and $\Pi$, constitute observe phase processing.
4. Generalized word- and phrase-level bindings are interpreted by the components $\Phi$ to form dialog states. Train, for example, is the dialog state of a training experience in the CRA.
5. The components of $\Gamma$ create action requests from bindings and dialog states. The maps $\Phi$ and $\Gamma$ constitute orient phase processing.
6. Scene bindings include user communications context. Context-sensitive plans are created by the component $\Sigma$ that evaluates action requests in the plan phase.
7. The decision phase processing consists of map $\Delta$ that maps plans and scene context to actions.
8. Finally, the map $E$ (consists of the effector components that change the PDA's internal states, change displays, synthesize speech, and transmit information on wireless networks using the SWR personalities.

**CRA Identifies Self, Owner, and Home Network**

The sets of entities in the world that are known to the CR are modeled graphically as rounded rectangles in Figure 14.12.

These include the self-grounded in the outside world (“self”), as well as its knowledge of the self as self (e.g., as “PDA”).
The critical entities are world, W, the PDA, and the PDA’s World Model, S. (In the CRA, S includes the orient phase data structures and processes.) Entities in the world include the differentiated entities “Own User” or owner, and “Home Network.”

The architecture requires that the PDA be able to identify these entities so that it may treat them differentially. Other networks, people, places, and things may be identified in support of the primary cognition functions, but the architecture does not depend on such a capability.

**CRA-Reinforced Hierarchical Sequences**

The data structures for perception include the reinforced hierarchical sequences words, phrases, dialogs, and scenes of the observe phase.

Within each of these sequences,

- the novel sequences represent the current stimulus–response cases of the cognitive behavior model.
- the known sequences represent the integrated knowledge of the cognitive behavior model. Known sequences may consist of a priori RXML statements embedded in the PDA or of knowledge acquired through independent ML.

The nearest sequence is the known sequence that is closest in some sense to the novel sequence. The World Model, W, consists primarily of bindings between a priori data structures and the current scene.

These associative structures are also associated with the observe phase.

Dialog states, action requests, plans, and actions are additional data structures needed for the observe, orient, plan, and act phases, respectively.

Each internal data structure maps to an RXML frame consisting of element (e.g., set or stimulus); model (e.g., embedded procedure, parameter values); content, typically a structure of elements terminating in either primitive concepts <concept/> (e.g., subset or response) or instance data; and associated resources.

Context is defined as the RXML URL or root from <Universe>, to include source, time, and place of the <Scene>.

**Behaviors in the CRA**

CRA entails three modes of behavior: waking, sleeping, and praying. Behavior that lasts for a specific time interval is called a behavioral epoch. The axiomatic relationships among these behaviors are expressed in the topological maps of Figure 14.13.

**Waking Behavior**

Waking behavior is optimized for real-time interaction with the user, isochronous control of SWR assets, and real-time sensing of the environment.

The conduct of the waking behavior is informally referred to as the awake-state, although it is not a specific system state, but a set of behaviors. Thus, referring to Figure 14.13, the awake-state cognition-actions (α) map the environment interactions to the current stimulus–response cases.
Sleeping and Dreaming Behaviors

Cognitive PDAs (CPDAs) detect conditions that permit or require sleep and dreaming.

For example, if the PDA predicts or becomes aware of a long epoch of low utilization (such as overnight hours), then the CPDA may autonomously initiate sleeping behavior.

Sleep occurs during planned inactivity, for example, to recharge batteries.

Dreaming behavior employs energy to retrospectively examine experience since the last period of sleep.

In the CRA, all sleep includes dreaming. In some situations, the CPDA may request permission to enter sleeping/dreaming behavior from the user (e.g., if predefined limits of aggregate experience are reached).

Prayer Behavior

Attempts to resolve unresolved conflicts via the mediation of the PDA’s home network may be called prayer behavior, referring the issue to a completely trusted source with substantially superior capabilities. The unresolved-conflicts list $\gamma$ is mapped $\lambda$ to RXML queries to the PDA’s home CN expressed in XML, OWL, KQML, RKRL, RXML, or a mix of declared knowledge types. Successful resolution maps network responses to integrated knowledge $\mu$. 

Figure 14.13: Cognitive behavior model consists of domains and topological maps

- $\alpha$: action cycle; $\delta$: incremental machine learning;
- $\beta$: nonincremental machine learning; $\gamma$: learning conflicts;
- $\lambda$: RKRL/KQML requests for external assistance;
- $\mu$: authoritative assistance; $\chi$: attempt to resolve problems.
Building the CRA on SDR Architectures

A CR is an SWR or SDR with flexible formal semantics-based entity-to-entity messaging via RXML and integrated ML of the self, the user, the RF environment, and the “situation.”

Review of SWR and SDR Principles

Hardware-defined radios such as the typical amplitude/frequency modulation (AM/FM) broadcast receiver convert radio to audio using such radio hardware as antennas, filters, analog demodulators, and the like.

SWR is the ideal digital radio in which the analog-to-digital converter (ADC) and digital-to-analog converter (DAC) convert digital signals to and from RF directly, and all RF channel modulation, demodulation, frequency translation, and filtering are accomplished digitally.

Figure 14.14 shows how SDR principles apply to a cellular radio-base station.

The ideal SWR would have essentially no RF conversion, just ADC/DAC blocks accessing the full RF spectrum available to the (wideband) antenna elements.

In this architecture, RF conversion can be a substantial system component, sometimes 60 percent of the cost of the hardware, and not amenable to cost improvements through Moore’s law.

The ideal SDR would access more like 2.5 GHz from, say 30 MHz to around 2.5 GHz, supporting all kinds of services in TV bands, police bands, air traffic control bands, and other bands.

![Diagram](image)

Figure 14.14: SWR principle applied to cellular-base station

The ideal SWR is not readily approached in many cases, so the SDR has comprised a sequence of practical steps from the baseband DSP of the 1990s toward the ideal SWR. As the economics of Moore’s law and of increasingly wideband RF and IF devices allow, implementations move upward and to the right in the SDR design space (Figure 14.16).
This space consists of the combination of digital access bandwidth and programmability.

Access bandwidth consists of ADC/DAC sampling rates converted by the Nyquist criterion and practice into effective bandwidth.

Programmability of the digital subsystems is defined by the ease with which logic and interconnect may be changed after deployment.

Application-specific integrated circuits (ASICs) cannot be changed at all, so the functions are “dedicated” in silicon.

Field-programmable gate arrays (FPGAs) can be changed in the field, but if the new function exceeds some performance parameter of the chip, which is not uncommon, then one must upgrade the hardware to change the function, just like with ASICs.

DSPs are typically easier or less expensive to program and are more efficient in power use than FPGAs. Memory limits and instruction set architecture (ISA) complexity can drive up costs of reprogramming the DSP.

Finally, general-purpose processors, particularly reduced instruction set computers (RISCs), are most cost-effective to change in the field. To characterize a multiprocessor, such as a cell phone with a CDMA-ASIC, DSP speech codec, and RISC microcontroller, weight the point in the design space by equivalent-processing capacity.
Radio Architecture

The discussion of the SWR design space contains the first elements of radio architecture.

It defines a mix of critical components for the radio. For SWR, the critical hardware components are the ADC, DAC, and processor suite.

The critical software components are the user interface; the networking software; the information security (INFOSEC) capability (hardware and/or software); the RF media access software, including the physical (PHY) layer modulator and demodulator (modem) and media access control (MAC); and any antenna-related software, such as antenna selection, beamforming, pointing, and the like.

INFOSEC consists of transmission security, such as the frequency-hopping spreading code selection, plus communications security encryption.

The SDR Forum defined a very simple, helpful model of radio in 1997, which is shown in Figure 14.17.

![Figure 14.17: SDR Forum (formerly MMITS) information transfer thread architecture](image)

The CR has to “know” about these functions, so this model is a good start because it shows both the relationships among the functions and the typical flow of signal transformations from analog RF to analog or (with SDR) digital modems, and on to other digital processing, including system control of which the user interface is a part.

The SCA

The US DoD developed the SCA for its Joint Tactical Radio System (JTRS) family of radios.

The SCA identifies the components and interfaces shown in Figure 14.18.

The APIs define access to the PHY layer, to the MAC layer, to the logical link control (LLC) layer, to security features, and to the input/output of the physical radio device.

The physical components consist of antennas and RF conversion hardware that are mostly analog and that typically lack the ability to declare or describe themselves to the system.
Most other SCA-compliant components are capable of describing themselves to the system to enable and facilitate plug-and-play among hardware and software components. In addition, the SCA embraces the portable operating system interface (POSIX) and CORBA.

![Diagram of SCA components](image)

**Figure 14.18: JTRS SCA Version 1.0 (© 2004 SDR Forum, used with permission).**

**Functions-Transforms Model of Radio**

The CRA uses a self-referential model of a wireless device, the functions-transforms model, to define the RKRL and to train the CRA.

In this model, illustrated in Figure 14.21, the radio knows about sources, source coding, networks, INFOSEC, and the collection of front-end services needed to access RF channels.

Its knowledge also extends to the idea of multiple channels and their characteristics (the channel set), and the radio part may have many alternative personalities at a given point in time.

Through evolution support, those alternatives change over time.

![Diagram of functions-transforms model](image)

**Figure 14.21: Functions-transforms model of a wireless node**
Architecture Migration: From SDR to AACR

Given the CRA and contemporary SDR architecture, one must address the transition of SDR through a phase of AACRs, toward the iCR.

As the complexities of handheld, wearable, and vehicular wireless systems increase, the likelihood that the user or network will have the skill necessary to do the optimal thing in any given circumstance is reduced.

Today’s cellular networks manage the complexity of individual wireless protocols for the user, but the emergence of multiband multimode AACR moves the burden for complexity management toward the PDA.

Likewise, the optimization of the choice of wireless service between the “free” home WLAN and the “for-sale” cellular equivalent moves the burden of radio-resource management from the network to the WPDA.

Radio Evolution toward the CRA

Various protocols have been proposed by which radio devices may share the radio spectrum.

The US FCC Part 15 rules permit low-power devices to operate in some bands. In 2003, a Report and Order (R&O) made unused TV spectrum available for low-power RF-LAN applications, making the manufacturer responsible for ensuring that the radios obey this simple constraint. DARPA’s XG program developed a language for expressing spectrum use policy.

Thus, one can envision a gradual evolution toward the CRA beginning initially with a minimal set of functions mutually agreeable among the growing community of AACR stakeholders. Subsequently, the introduction of new services will drive the introduction of new capabilities and additional APIs, perhaps informed by the CRA.
UNIT-IV
PART- A Two Marks

1. Define AACRs.
   
   Aware, Adaptive, and Cognitive Radios (AACRs), AACR consists of six functional components: user SP, environment, effectors, SDR, sys apps, and cognition. The capabilities required for an AACR node to be a cognitive entity are to sense, perceive, orient, plan, decide, act, and learn.

2. What is RXML?
   
   Radio eXtensible Markup Language, <Radio> defines “the domain of natural and artificial knowledge and skill having to do with the creation, propagation and reception of radio signals from natural and artificial sources.”

3. Draw the block shows functional components of cognitive radio architecture.

![Minimal AACR node architecture](image)

4. What are the functional components of CRA?
   
   1. The user sensory perception (SP), which includes haptic, acoustic, and video sensing and perception functions.
   2. The local environment sensors (location, temperature, accelerometer, compass, etc.).
   3. The system applications (sys apps) media-independent services such as playing a network game.
   4. The SDR functions which include RF sensing and SDR applications.
   5. The cognition functions (symbol grounding for system control, planning, and learning).
   6. The local effector functions (speech synthesis, text, graphics, and multimedia displays).

5. What are the computational intelligence aspects of CR?
   
   Three computational intelligence aspects of CR:
   1. Radio knowledge—RXML:RF
   2. User knowledge—RXML:User
   3. The capacity to learn

   
   The cognition component must to assess, manage, and control all of its own resources, including validating downloads.
   
   Thus, in addition to <RF> and <User> domains, RXML must describe the <Self/>, defining the AACR architecture to the AACR itself in RXML.
7. What is watchdog timer?

A **watchdog timer** (WDT) is a hardware **timer** that automatically generates a system reset if the main program neglects to periodically service it. It is often used to automatically reset an embedded device that hangs because of a software or hardware fault.

Without the reliable watchdog timer in the architecture computer programmers would build CRs that would crash in extremely unpredictable ways as their adaptation algorithms got trapped in unpredictable unbounded self-referential loops.

8. What is the role of sensory functions in cognitive radio?

*The sensory functions of radio* entail those hardware and/or software capabilities that enable a radio to measure features of a sensory domain. Sensory domains include anything that can be sensed, such as audio, video, vibration, temperature, time, power, fuel level, ambient light level, sun angle (e.g., through polarization), barometric pressure, smell, and anything else imaginable.

9. What is Cognition Cycle?

The way of implementing the cognition technique in a communication system includes a primary cycle of cognition such as, Observe, Orient, Plan, Decide, Act (OOPDA) Loop; Learning, Planning and sensing the outside world are the crucial phases of the OOPDA loop.


11. Write short note on orient phase of cognition.

Orient phase is part of the cognition component.

The orient phase determines the significance of an observation, by binding the observation to a previously known set of stimuli of a —scene.

The orient phase contains the internal data structures that constitute the equivalent of the short-term memory (STM) that people use to engage in a dialog without necessarily remembering everything with the same degree of long-term memory (LTM).
12. What is binding in Orient Phase?
Binding occurs when there is a nearly exact match between a current stimulus and a prior experience and very general criteria for applying the prior experience to the current situation are met.

Binding also determines the priority associated with the stimuli. Better binding yields higher priority for autonomous learning, whereas less-effective binding yields lower priority for the incipient plan.

13. Write short note on Plan in cognition cycle.
A stimulus may be associated with a simple plan as a function of planning parameters with a simple planning system.
Open source planning tools enable the embedding of planning subsystems into the CRA, enhancing the plan component.

14. Write short note on learning in cognition cycle.
Learning is a function of perception, observations, decisions, and actions.
Initial learning is mediated by the observe phase perception hierarchy in which all SP are continuously matched against all prior stimuli to continually count occurrences and to remember time since the last occurrence of the stimuli from primitives to aggregates.

Learning also occurs through the introduction of new internal models in response to existing models and case-based reasoning (CBR) bindings.

15. Write short note on the standard inference hierarchy
The phases of inference from observation to action show the flow of inference, a top-down view of how cognition is implemented algorithmically.

The inference hierarchy is the part of the algorithm architecture that organizes the data structures. An illustrative inference hierarchy includes layers from atomic stimuli at the bottom to information clusters that define action contexts.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Level of Abstraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context Cluster</td>
<td>Scenes in a Play, Session</td>
</tr>
<tr>
<td>Sequence Clusters</td>
<td>Dialogs, Paragraphs, Protocol</td>
</tr>
<tr>
<td>Basic Sequences</td>
<td>Phrases, Video Clip, Message</td>
</tr>
<tr>
<td>Primitive Sequences</td>
<td>Words, Token, Image</td>
</tr>
<tr>
<td>Atomic Symbols</td>
<td>Raw Data, Phoneme, Pixel</td>
</tr>
<tr>
<td>Atomic Stimuli</td>
<td>External Phenomena</td>
</tr>
</tbody>
</table>

16. What is Atomic Stimuli?
Atomic stimuli originate in the external environment and are sensed and preprocessed by the sensory subsystems, which include sensors of the RF environment (e.g., radio receiver and related data and information processing) and of the local physical environment, including acoustic, video, and location sensors.

17. What is Atomic Symbols?
Atomic symbols are the elementary stimuli extracted from the atomic stimuli.
Atomic symbols may result from a simple noise-riding threshold algorithm, such as the squelch circuit in RF that differentiates signal from noise.

18. What is context cluster?
A scene is a context cluster, a multidimensional space–time–frequency association, such as a discussion of a baseball game in the living room on a Sunday afternoon. Such clusters may be inferred from unsupervised Machine Learning.

Cognition functions are implemented via cognition elements consisting of data structures, processes, and flows, which may be modeled as topological maps over the abstract domains identified in figure below.

20. Write a note on CRA behavior.
CRA entails three modes of behavior: waking, sleeping, and praying. Behavior that lasts for a specific time interval is called a behavioral epoch.

21. Write a short note on waking behavior.
Waking behavior is optimized for real-time interaction with the user, isochronous control of SWR assets, and real-time sensing of the environment.
The conduct of the waking behavior is informally referred to as the awake-state, although it is not a specific system state, but a set of behaviors.

22. Write a short note on sleeping and dreaming behaviors.
Cognitive PDAs (CPDAs) detect conditions that permit or require sleep and dreaming.

For example, if the PDA predicts or becomes aware of a long epoch of low utilization (such as overnight hours), then the CPDA may autonomously initiate sleeping behavior.

Sleep occurs during planned inactivity, for example, to recharge batteries.
Dreaming behavior employs energy to retrospectively examine experience since the last period of sleep.

23. Write a short note on Prayer Behavior.

Attempts to resolve unresolved conflicts via the mediation of the PDA’s home network may be called prayer behavior, referring the issue to a completely trusted source with substantially superior capabilities.

24. Draw the SDR Design Space.

25. Draw the cognitive behavior model.
Part- B
16 Mark

1. Explain the functions and components of cognitive radio with neat diagram.
2. Explain the about the cognition cycle with necessary diagrams
3. Explain about the cognition components
4. Explain about the design rules for functional component interfaces.
5. Explain about the Inference Hierarchy and its phases.
6. Explain the Cognitive Radio Architecture Maps with neat diagrams
7. Explain the concept of building the CRA on SDR architectures and its evolution from hardware radio briefly.