باسمه‌تعالی

سمینار درس رادیو نرم‌افزاری با موضوع:

مقدمه‌ای بر

Software Communication Architecture (SCA)

استاد درس: آقای دکتر امیدی

ارائه‌دهنده: م. هلاجیان

دانشکده برق و کامپیوتر دانشگاه صنعتی اصفهان

تابستان ۸۸

• مقدمه:

- معماری نرم‌افزار: چارچوبی که ساختار نرم‌افزار شامل اجزاء نرم‌افزار و روابط آن اجزاء را مشخص می‌کند

- رویکرد به سمت رادیو نرم‌افزاری ⟵ هر توسعه دهنده یک معماری دلخواه برای نرم‌افزار

- وجود یک ساختار استاندارد: سازگاری سیستم‌ها ، انعطاف بیش‌تر ، مدیریت ساده‌تر

- پروژه‌ی JTRS (Joint Tactical Radio Systems):

  - یکی از بزرگ‌ترین پروژه‌های رادیو نرم‌افزاری
  - شروع از اواسط دهه‌ی ۱۹۹۰ در ارتش امریکا
  - بنیان نسل آینده‌ی رادیوهای تاکتیکی ارتش امریکا
  - یکی از محصولات این پروژه ⟵ معماری با عنوان SCA
    ⟵ قبل اعمال به تمام انواع رادیو نرم‌افزاری

• SCA چیست؟

- SCA (Software Communication Architecture): یک معماری خاص.

- مورد پشتیبانی OMG و SDRF برای تبدیل شدن به یک استاندارد تجاری بین‌المللی

  ← مطرح‌ترین معماری برای نرم‌افزار رادیو نرم‌افزاری

-SCA: تجرید بین اجزاء مختلف سخت‌افزاری و نرم‌افزاری:

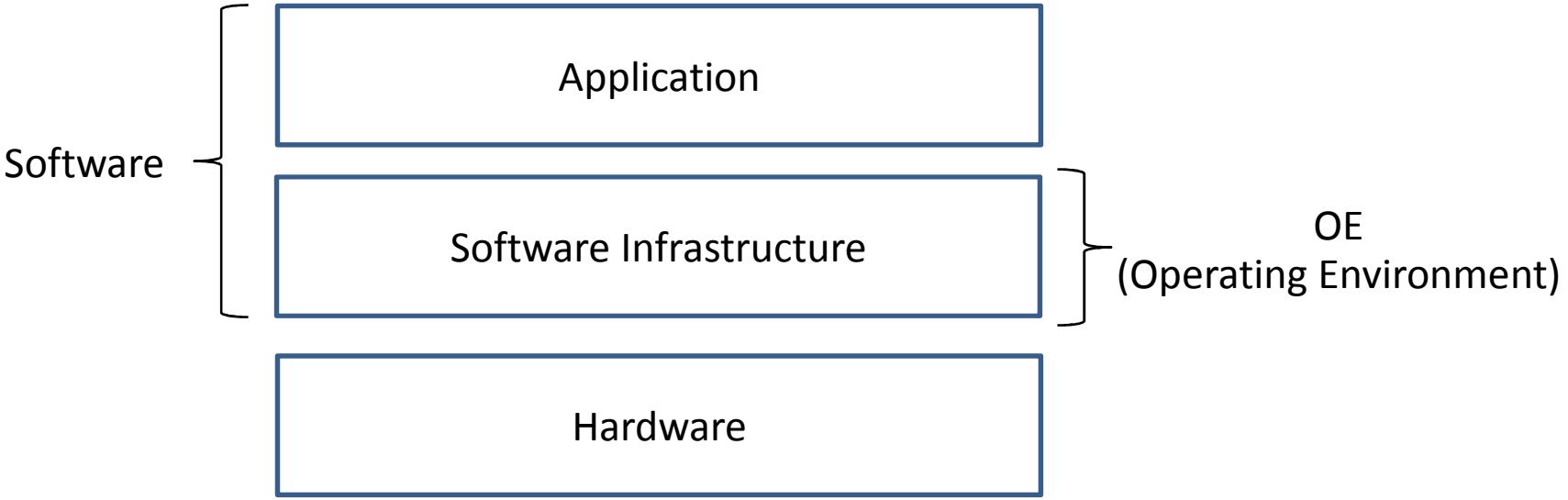
- ارتقاء یک قسمت از سیستم با ظهور یک تکنولوژی جدید بدون تغییر سایر قسمت‌ها
- بهره‌مندی توسعه‌دهندگان مختلف از نتیجه‌ی کار یک‌دیگر
- کاربرد پیاده‌سازی شده بر روی یک سیستم با تغییرات جزئی قابل انتقال به سیستم دیگر

- SCA تنها چارچوب و قیود را تعیین می‌کند و توضیح تکنیکی برای پیاده‌سازی ارائه نمی‌کند.
  و اجباری به استفاده از یک تکنولوژی خاص ایجاد نمی‌کند.

- جزئیات SCA با استفاده از نمادهای گرافیکی تعریف شده در زبان مدل‌سازی UML (Unified Modeling Language) توصیف می‌شود.

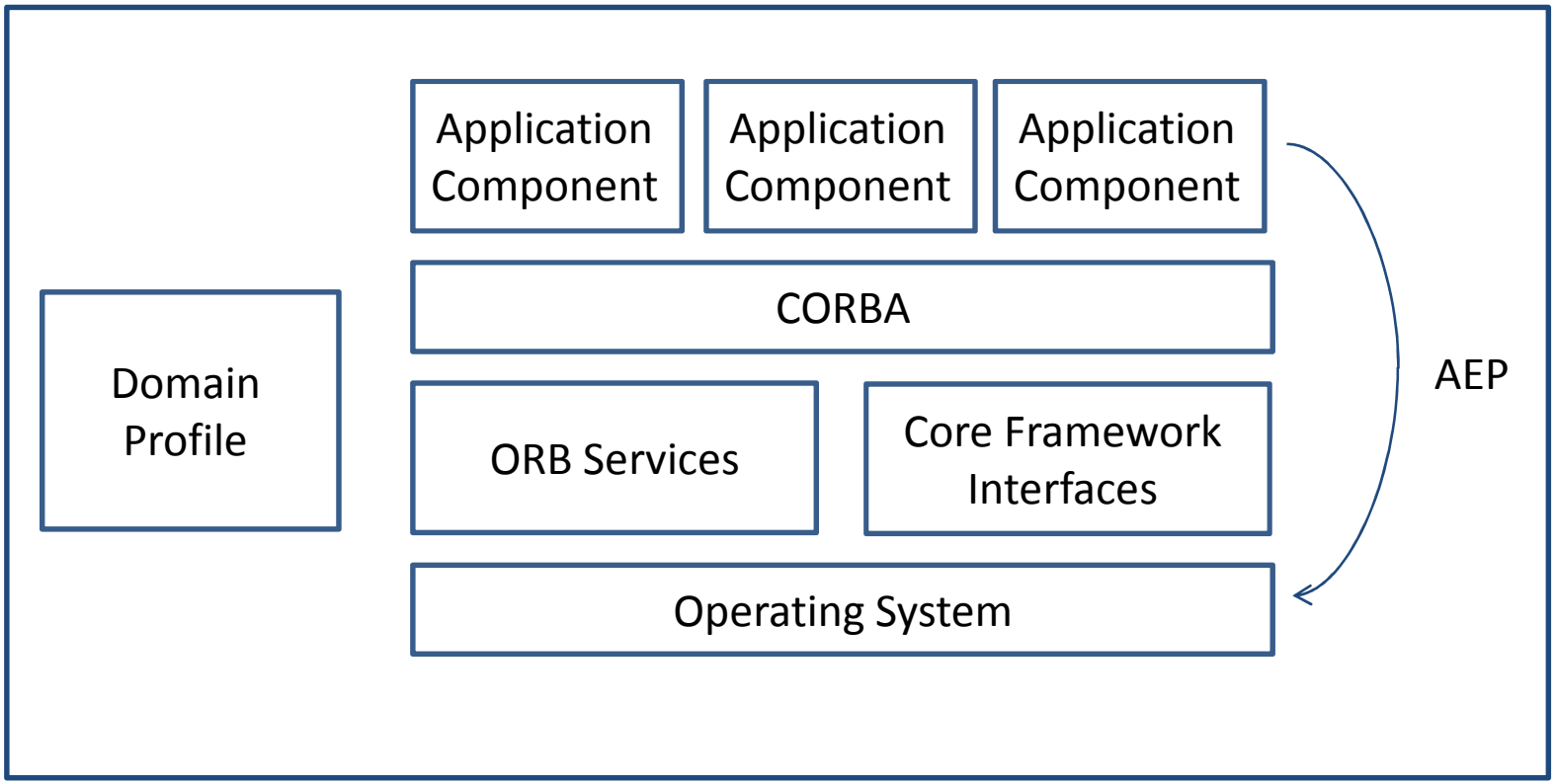
• ساختار SCA:

- اجزاء مختلف یک رادیوی نرم‌افزاری:



- SCA، ملاحظاتی را درباره‌ی Software Infrastructure و تا حدودی درباره‌ی Applications، مطرح می‌کند.

- ساختار SCA:


Application Component
Application Component
Application Component
CORBA
Domain Profile
ORB Services
Core Framework Interfaces
Operating System
AEP

• Application Components:

- SCA، رادیو را به عنوان یک ساختار پردازش توزیع یافته در نظر می‌گیرد.

Application ها ⟵ مجموعه‌ای از کامپوننت‌های نرم‌افزاری که بر روی اجزای پردازشی سیستم بارگذاری می‌شوند.

- کامپوننت‌های Application: سازگار با CORBA ⟵ ارتباط این کامپوننت‌ها با یکدیگر و با CF از طریق استاندارد CORBA

- منابع در دسترس برای برنامه‌نویس کامپوننت App
  - به اینترفیس‌های CoreFramework از طریق CORBA
  - به سیستم عامل از طریق پروفایل AEP

• CORBA (Common Object Request Broker Architecture):

- استانداردی برای نرم‌افزارهای کامپوننتی:
- تعریف پروتکل‌های رابطه‌ای برای ارتباط اجزاء نرم‌افزاری نوشته شده به زبان‌های متفاوت.
- بسته‌بندی کدهای نوشته شده به زبان‌های دیگر و افزودن الحاقاتی به آن.
- ارتباط کامپوننت‌ها، بدون اطلاع از موقعیت مکانی یک‌دیگر و جزئیات پیاده‌سازی.
- ابزار اجرایی CORBA: زبان IDL (Interface Definition Language).

- این کامپوننت‌ها، نه به طور مستقیم، بلکه به واسطه‌ی سرویس‌های خاصی تحت عنوان ORB (مثل Name Service، Event Service و ...) با یکدیگر ارتباط برقرار می‌کنند.

• Core Framework Interfaces:

- Interface ؟

- Core Framework : مجموعه‌ی ضروری از اینترفیس‌های نرم‌افزاری

⟵ ایجاد یک تجرید نسبی بین کامپوننت‌های Application
با سایر اجزاء نرم‌افزاری و سخت‌افزاری سیستم.

- انواع اینترفیس‌های تعریف شده در Core Framework:

۱- Base Application Int. ⟵ اینترفیس‌های برای کنترل و مدیریت کامپوننت‌های نرم‌افزاری.

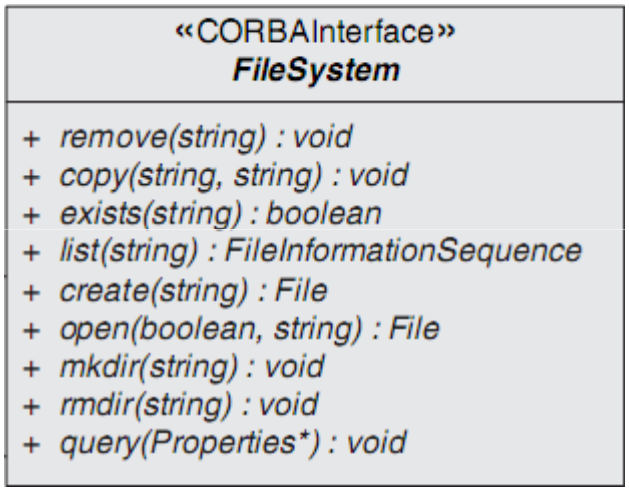
۲- Base Device Int. ⟵ اینترفیس‌هایی برای کنترل و مدیریت قطعات سخت‌افزاری درون سیستم

۳- Framework Control Int. ⟵ اینترفیس‌هایی برای نصب، پیکربندی و مدیریت کاربردها.

۴- Framework Services Int. ⟵ ایجاد یک فایل سیستم متحد بین بسترهای پردازشی مختلف.

- **Core Framework Interfaces:**

یک نمونه از اینترفیس‌های مطرح شده در Framework Services Int.:



۳- Framework Control Int. ⟵ اینترفیس‌هایی برای نصب، پیکربندی و مدیریت کاربردها.

۴- Framework Services Int. ⟵ ایجاد یک فایل سیستم متحد بین بسترهای پردازشی مختلف.

• Operating System:

- سیستم عاملی است که بر روی بسترهای پردازشی سیستم رادیویی اجرا شده و توابع Real-Time مورد نیاز برای اجزای مختلف نرم‌افزاری و قطعات سخت‌افزاری را فراهم می‌کند:

مثال: - بوت‌کردن پروسسورها.

- پشتیبانی از پردازش چندتاری.

- پشتیبانی از I/O ها.

- و سایر قابلیت‌های چندمنظوره‌ای که برای کاربردهای Real-Time مورد نیاز است.

- یک سیستم رادیویی ممکن است یک یا چند سیستم‌عامل داشته باشد. و یا اصلا فاقد چنین سیستم عاملی باشد.

- مثال: OS مورد استفاده در SCARI ⟵ هسته‌ی Linux !

• Operating System:

- دسترسی محدود کامپوننت‌های Application به سیستم عامل:

- علت محدودیت دسترسی: به منظور حفظ قابلیت حمل‌پذیری کاربردها.

- POSIX:
  - یکی از استانداردهای IEEE
  - مجموعه‌ای از رابط‌های دسترسی به سیستم عامل UNIX را مشخص می‌کند.
  - قابل تعمیم به تمام انواع سیستم عامل‌ها نیز هست.

- AEP: مجموعه‌ی کمینه‌ای از رابط‌های تعریف شده در POSIX که ارتباط کامپوننت‌های کاربرد محدود به این رابط‌هاست.

- **Domain Profile**:

- توصیف اجزاء مختلف سخت‌افزاری و نرم‌افزاری سیستم در قالب فایل‌های XML

  ← مثل: قابلیت‌ها، وابستگی‌ها، موقعیت مکانی قطعات و ...

- این فایل‌ها مورد استفاده‌ی اجزای مختلف سیستم قرار می‌گیرند ← خواندن / نوشتن

- مثال: سرویس‌های ORB از این اطلاعات برای برقراری ارتباط اجزای سازگار با CORBA استفاده می‌کنند.

- مثال: هنگام نصب یک Application ← تطابق نیازهای کاربرد با امکانات موجود از طریق مقایسه‌ی این فایل‌های XML.

- فایده‌ی Domain Profile: تجمیع اطلاعات مورد نیاز سیستم در

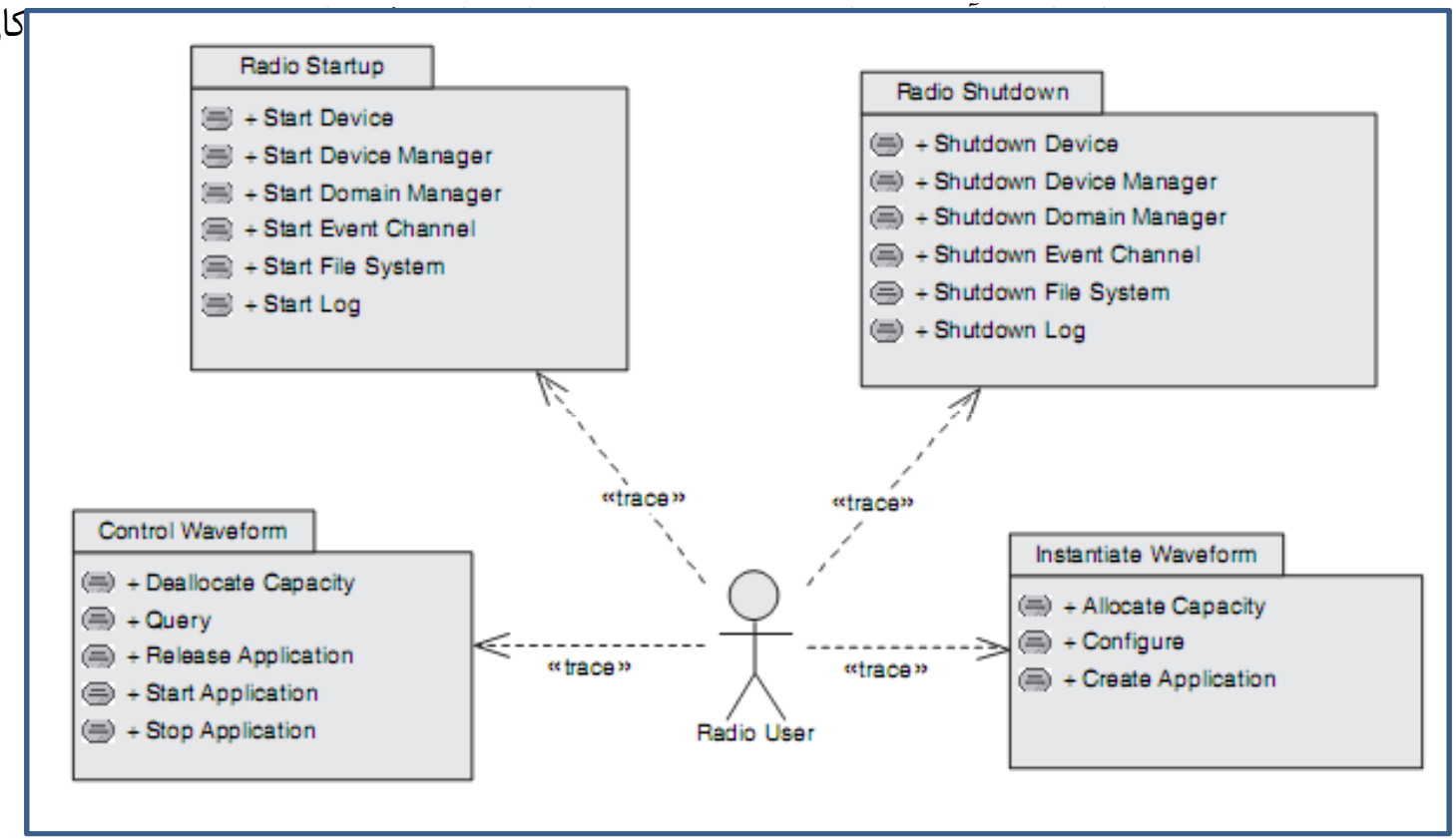
- **Operational Scenarios**:

- سناریوهایی که کاربر از طریق آن‌ها با رادیو تعامل می‌کند. و رادیو از رهگذر این سناریوها مقاصد کاربر را برآورده می‌کند.

- دو نوع کاربر:
  - Common User

- **Operational Scenarios:**

کاربر



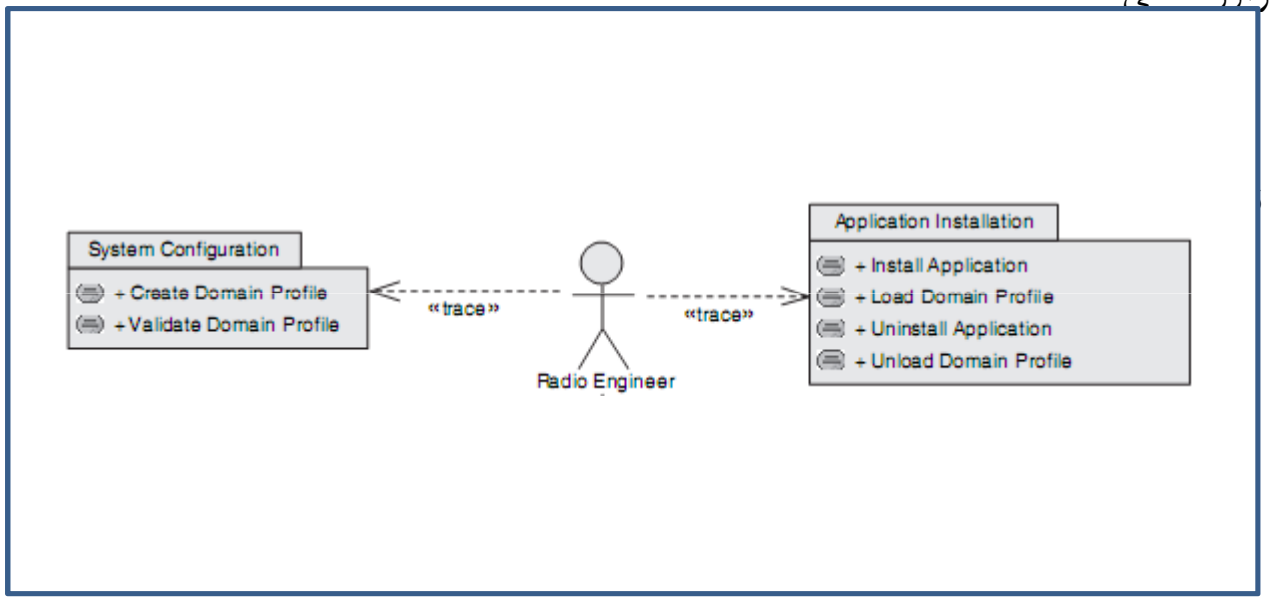
- **Operational Scenarios**:

- سناریوهایی که کاربر از طریق آن‌ها با رادیو تعامل می‌کند. و رادیو از رهگذر این سناریوها مقاصد کاربر را برآورده می‌کند.

- دو نوع کاربر:
  - Common User
  - Radio Engineer

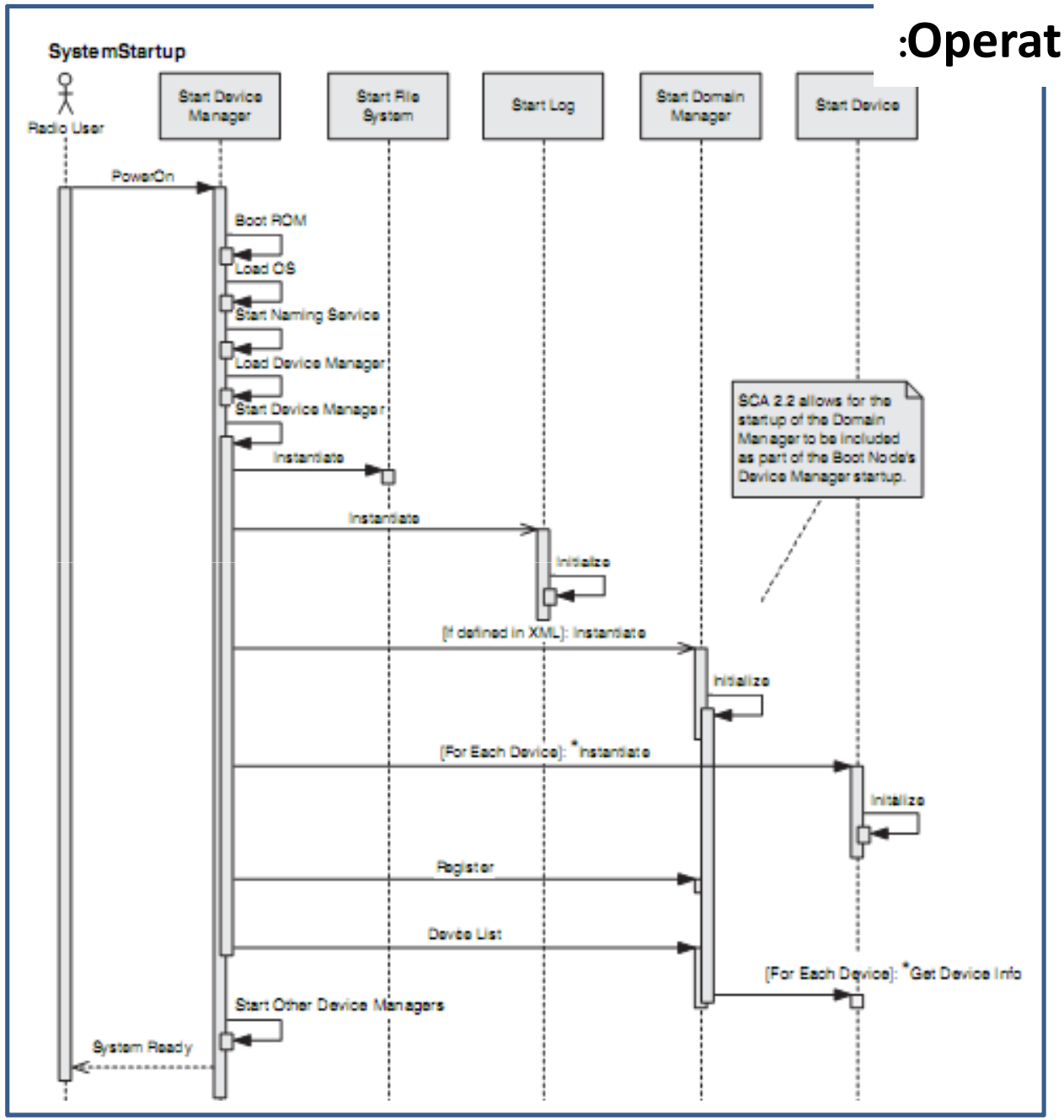
- **Operational Scenarios:**

- سناریوهایی که کاربر از طریق آن‌ها با رادیو تعامل می‌کند. و رادیو از رهگذر این سناریوها مقاصد کاربر را برآورده می‌کند.



- دو

Operational Scenarios :•



- مثال: System Startup

Operational Scenario

- **پیشنهادات:**

  - **دو رویکرد برای مطالعه‌ی SCA:**

رویکرد اول:

- مطالعه‌ی پیش‌نیازهایی که معرفی SCA بر آن‌ها استوار است. مثل: CORBA، IDL، UML.

- مطالعه‌ی جزئیات استاندارد SCA.

رویکرد دوم:

- مطالعه‌ی پروژه‌های رادیو نرم‌افزاری اپن سورس مبتنی بر SCA.

مثال: OSSIE، SCARI