GSM and IN Architecture

a common component: TCAP

Raimo.Kantola@netlab.hut.fi

GSM system consists of sub-systems

- Mobile Station
- Mobile Equipment
- Subscriber Identity Module
- Base Station Subsystem
- HLR belongs to both CS and PS domains

Main differences cmp to wire-line networks
- air interface for the subscribers
- mobility and roaming of users
NB: the whole system is digital incl the ME.
The original GSM architecture

- **HLR/AC/EIR**
  - HLR - Home Location Register (kotirekisteri)
  - AC - Authentication Center (Varmennekeskus)
  - EIR - Equipment Identity Register (laiterekisteri)

- **MSC**
  - Mobile Switching Center (matkapuhelinkeskus)
  - GMSC – Gateway MSC

- **VLR**
  - Visitor Location Register (vierailijarekisteri)

- **BSC**
  - Base Station Controller (tukiasemaohjain)
  - BTS - Base Transceiver Station (tukiasema)

**CS Core interfaces are**

- **VLR**
  - Visitor Location Register
  - NB: Additionally on MSC to MSC interface ISUP is used.

- **MSC**
  - Mobile Switching Center
  - NB: MSC+VLR always in the same node
  - +I - interface: MS - HLR (MS-MSC/VLR-HLR)

- **HLR**
  - Home Location Register

**Base Station sub-system**

NB: All CS Core interfaces: B-G conform to the MAP protocol
Business boundaries in GSM

✓ Vendors and operators
  › An operator can not just assume that all vendors conform to all interfaces that are found in GSM specs
  › A-interface is firmly adhered to by all vendors: an operator can buy the cellular radio network from one vendor and the rest from another
  › Also the Packet core is rather independent of the circuit network core – again two different vendors is a feasible alternative
  › Advice: it is a good idea to buy HLR and MSC from the same vendor – if not two vendors may introduce features in a different order and the operator instead of getting the superset of features is getting the intersection of features.

✓ Mobile Virtual operators (MVO) of different types
  › An MVO may but is not forced to have its own HLR
  › The MVO HLR may need to work with MSC from a different vendor – not impossible

CAMEL adapts the IN technology to GSM

✓ CAMEL - Customized Application for Mobile network Enhanced Logic
✓ The goal is the capability of providing the home network services to visiting subscribers
✓ CAP - CAMEL Application Part is a subset of ETSI CoreINAP
  › phases (Capability Sets) 1…4 are ready
IN is a way of implementing services in nodes separate from exchanges

INAP = IN Application Part
= main protocol
CCF – Call Control Function
SSF - Service Switching Function
maintains call state with CCF
SCF - Service Control Function
implements service logic
SRF - Special Resource Function
processes in-band signals
SDF - Service Data Function
is a database
SCE - Service Creation Environment
for creating new service logic
SMP - Service Management Point
implements mgt functions

Network of STP nodes

Features of the IN architecture ...

✓ BCSM - Basic Call State Model is a standardized state machine in SSP - couples/ de-couples IN service logic from connection resources
✓ BCSM states (detection points) can be programmed to trigger queries on conditions to an SCF concerning a certain call
✓ BCSM architectural issue is that a call is also a service and therefore the architecture is service dependent
✓ INAP messages are independent of voice channel connections
Phase 1 CAMEL architecture

MS originated CAMEL call

A - MSC gets the CAMEL service info from the VLR concerning the A subscriber, sees an active CAMEL service and hands the call to gsmSSF. gsmSSF queries gsmSCF:lle (service key, A-nr, B-nr, IMSI, location...)

B - gsmSCF can for example do a number translation

C - MSC sets up a call using the received info

NB: The the service for the visiting subscriber is executed cooperatively by the visited network (MSC/VLR) and the home network! → Home network is not autonomous in providing the service!
Mobile terminated CAMEL call

<table>
<thead>
<tr>
<th>PSTN</th>
<th>GMSC + gsmSSF</th>
<th>HLR</th>
<th>gsmSCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISUP-IAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISUP-IAM</td>
<td></td>
<td>MAP-SendRoutingInfo</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAP-SRInfoResult</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>CAP-InitialDP</td>
<td>CAP-Connect</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A - GMSC queries HLR of the location of the MS. HLR sends the terminating CAMEL service data of the subscriber.
B - GMSC hands the call to gsmSSF, which queries gsmSCF
   gsmSCF returns C-number that is used for routeing the call
C - GMSC sets up the call to C-number. If needed, GMSC can first do a new HLR query.

An SCF can interrogate HLR at any time

<table>
<thead>
<tr>
<th>MSC</th>
<th>CAP</th>
<th>gsmSCF</th>
<th>HLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP/L</td>
<td></td>
<td>MAP/J</td>
<td>AnyTimeInterrogation</td>
</tr>
</tbody>
</table>
IN+GSM integration based on CAMEL is a step towards 3G

- CAPv1 supports only 7 operations
- CAPv1 call model has only a few triggering points (TDP - trigger detection point)
- CAPv2 has 22 operations
- Still no triggering for Short Messages
- CAMEL compatible equipment is in use in many networks

Need to separate application logic states from communication states is common to IN and MAP

- IN application is concerned with the end user service, its implementation may be broken into several modules each with its own communication needs between SCP and SSP.
- HLR and MSC may be discussing about a handover, VLR update etc in the same context – each part of MAP has its own communication needs.
- It makes sense to have a common component between the application and SCCP that will provide services friendly for applications and take care of communicating with the other part of the application in a remote node (HLR, VLR, MSC, SCP, SSP etc)
TCAP - Transaction Capabilities Application Part is used by

- Mobile services (roaming and mobility management)
- Intelligent Network services
- Services that are independent of voice circuits (look-ahead ...)
- O&M applications
- etc

TCAP provides generic services supporting the execution of distributed transactions. Parties in the transactions can be exchanges, service nodes, data bases etc.

TCAP offers a way to implement services that are independent of network resources.

Summary of course scope
TCAP has two sub-layers

Component sub-layer: data units of the application protocol, requests and responses, dialogues: application context

Transaction sub-layer: message exchange between parties, optionally dialogues between parties.

TCAP has a lot of similarity with ROSE (Remote Operation Service Element) and ACSE (Association Control Service Element). ROSE and ACSE are OSI layer 7 services.

Transactions are sequences of events that allow to read or write some data entry or entries in a remote network node.

A TCAP use case

**BEGIN (OTID = x)**

**CONTINUE (OTID = y, DTID = x)**

**CONTINUE (OTID = x, DTID = y)**

**CONTINUE (OTID = y, DTID = x)**

**END (OTID = y)**

Begin begins a dialogue

During the dialogue Continue - messages are sent in both directions.

End-message closes the dialogue.

OTID - identifies the dialogue/for the sender of the transaction.

DTID - identifies dialogue/for the object of the transaction.
**TCAP supports four operation types**

- Class 1 - Both success and failure are reported.
- Class 2 - Only failures are reported.
- Class 3 - Only success is reported.
- Class 4 - Nothing is reported.

An operation is identified by the Invoke-Id - identifier.

Indication (ind) is associated with the request (req) based on the Invoke-id.

A user may have many ongoing active operations simultaneously.

TCAP is a purely end-to-end function. There may be many intermediate nodes in the CCS7 network that do not touch TCAP.

---

**Operations are identified and chained using the Invoke-Id**

- Operation is identified by the Invoke-Id.
- Indication (ind) is associated with the request (req) based on the Invoke-id.
- The Response can be a new operation request that is chained to the previous operation request using a link-identifier.
- A user may have many simultaneous operations.
The result of an operation sent to a remote system can be

- Result: Operation succeeded.
  - The result can also be segmented (chained)
- Error: Operation failed.
- Reject: Execution of the operation is not possible.
- Before sending the result, the remote system can send an arbitrary number of linked operations.

Non-structured dialogue transfers one or more components

- TC-user can send many components in Class 4 operations by a UNIDIRECTIONAL message.
- Components with the same dialogue-id can be sent in one message.
- Control over sequencing of operations is left to the application.

Components are delivered in the same order they were submitted!
A Structured dialogue has a beginning, information transfer, ending or abort

- Begin causes a transaction identifier to be reserved.
- The remote system can either continue the transaction or close it.
- Continue - messages are exchanged in a full-duplex mode.
- Closing options:
  - based on pre-arrangement independently
  - normally by the End-message or “abnormally” by an Abort message

The Component sub-layer is split into dialogue handling and component handling

Component primitives
- TC-Invoke (ind, req)
- TC-Result-L (ind, req)
- TC-Result-NL (ind, req)
- TC-U-Abort (ind, req)
- TC-P-Abort (ind)
- TC-L-Cancel (ind)
- TC-U-Cancel (req)
- TC-R-Reject (ind)
- TC-L-Reject (ind)
- TC-U-Reject (ind, req)

Dialogue primitives
- TC-Notice (ind)
- TC-UNI (ind, req)
- TC-Begin (ind, req)
- TC-Continue (ind, req)
- TC-End (ind,req)
- TC-U-Abort (ind, req)
- TC-P-Abort (ind)

Component sub-layer

Dialogue Handling

Component Handling
- Component coordinator
- Invocation
- State-machine
Component handling primitives are

TC_INVOKE - Invocation of an operation which may be linked to another operation

TC_RESULT_L - Only result or last part of segmented result of a successful operation

TC_RESULT_NL - non-last part of segmented result

TC_U_ERROR - reply to a previously invoked op that failed

TC_L_CANCEL - informs user of local timeout

TC_U_CANCEL - Causes local termination of op on TC_user request

TC_L_REJECT - local reject by Component sub-layer to TC_user

TC_R_REJECT - remote reject by remote component sub-layer

TC_U_REJECT - Rejection by TC_user indicating malformation

Transaction sub-layer handles the interfacing to the network layer

TCAP can use all address mechanisms supported by SCCP.

To the peer entity

UNIDIRECTIONAL
BEGIN
CONTINUE
END
ABORT

Transaction Coordinator

Transaction State-Machine

Network layer (SCCP)
State transition Diagram for Class 1 Operations

Most important users of TCAP are..

INAP  CAP  MAP  BSSAP  ISUP  TUP  MUP  HUP

IN  GSM  ISDN  PSTN  NMT

TC

SCCP

MTP levels

Level 3 - Signaling network (MTP3)
Level 2 - Signaling link (MTP2)
Level 1 - data link (MTP1)
Summary: TCAP added value is

- Decoupling the actions and states of an application from communication states for managing the flow of information with the remote end
- Takes care of managing the communication with the peer – lets the application concentrate on essential matters
  - four classes of service
  - report on success tells the application that the remote end has done its job for sure
  - report on failures speeds up recovery (but an application can not really rely on getting the report on every failure!)
  - or alternatively can let the application take care of all acknowledgements