From Anchor Model
to Relational Database

L. Rönnbäck
lars.ronnback@resight.se
Resight, Kungsholms Strand 179, 112 48 Stockholm

O. Regardt
olle.regardt@teracom.se
Teracom, Kaknästornet, 102 52 Stockholm

M. Bergholtz, P. Johannesson, P. Wohed
maria@dsv.su.se, pajo@dsv.su.se, petia@dsv.su.se
DSV, Stockholm University, Sweden

September 15, 2010

In this paper the rules for translating an anchor schema into a relational database schema are presented. Throughout the paper an example of the anchor model from Figure 1 is used.

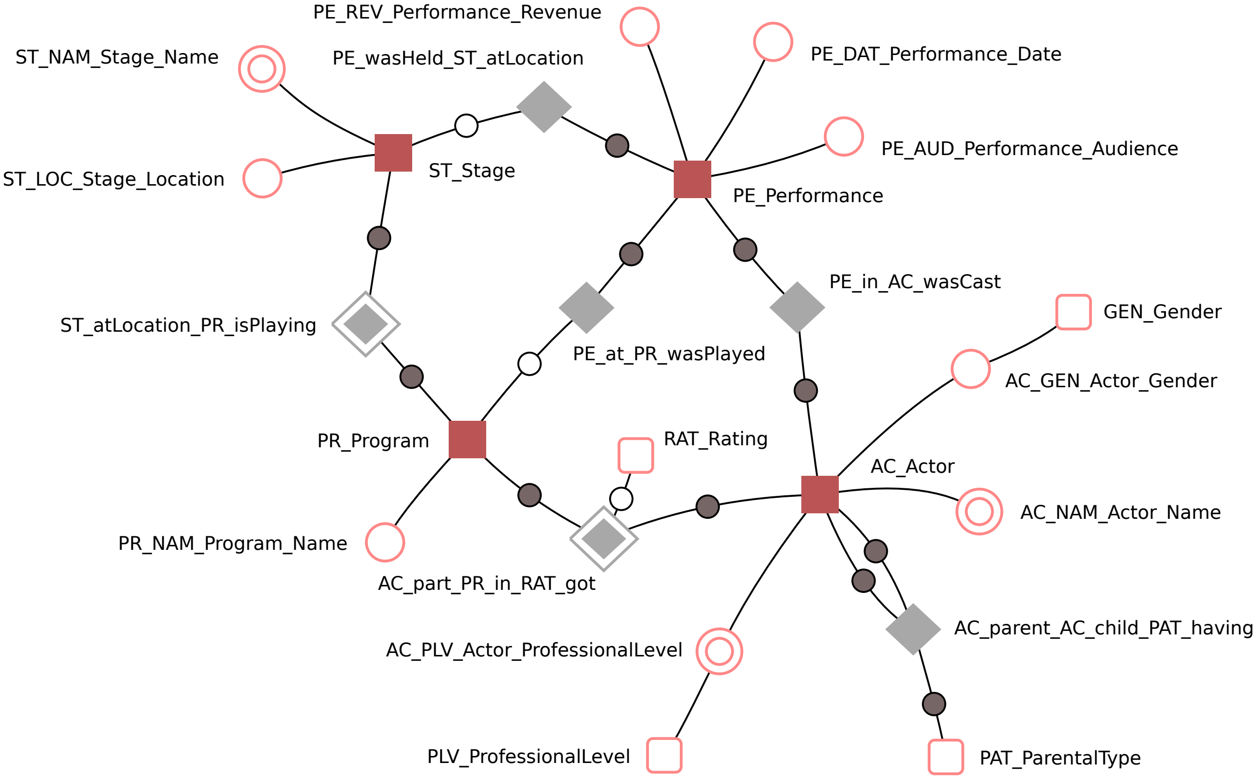

Figure 1: An anchor model

An anchor schema can be transformed into a relational database schema through the translation rules presented below. For each anchor, knot, attribute, and tie in an anchor schema there exists a translation rule that maps it into a relational database construct such as a relation schema (table definition) or column.

The names used in the relational database are derived from the naming convention which can be found on . The highly decomposed relational database schemas that result from anchor schemas facilitate traceability through metadata, capturing information such as creator, source, and time of creation. Although important, metadata is not discussed further since its use does not differ from that of other modeling techniques.

Anchors

An anchor A is mapped to a relation schema $A(S)$, where $S = N_{\mathbb{I}}(A)$. The domain of S is $\mathbb{I}$. S is the primary key of the relation schema $A(S)$.

AC_*Actor*

AC_ID *(PK)*
#4711 #4712 #4713
INTEGER

Figure 2: Example rows from the anchor AC_*Actor*.

For example the anchor AC_*Actor* with INTEGER as domain, see Figure 2, corresponds to the relation schema AC_*Actor*(AC_ID).

Knots

A knot K with range $\mathbb{D}$ is mapped to a relation schema $K(I, V)$, where $I = N_{\mathbb{I}}(K)$ and $V = N_{\mathbb{D}}(K)$. The domain of I is $\mathbb{I}$, and the domain of V is $\mathbb{D}$. I is the primary key.

GEN_*Gender*

GEN_ID *(PK)*	GEN_*Gender*
#0	'Male'
#1	'Female'
BIT	STRING

Figure 3: Example rows from the knot GEN_*Gender*.

For example the knot GEN_*Gender* with STRING as range and an extension over BIT × STRING, see Figure 3, corresponds to the relation schema GEN_*Gender*(GEN_ID, GEN_*Gender*).

Static Attributes

A static attribute B_S with domain A and range $\mathbb{D}$ is mapped to a relation schema $B(I, V)$, where $I = N_{\mathbb{I}_A}(A)$ and $V = N_{\mathbb{D}}(B)$. The domain of I is $\mathbb{I}$ and the domain of V is $\mathbb{D}$. The column I is a primary key of B and a foreign key with respect to the relation schema $A(S)$.

For example the attribute PE_DAT_*Performance_Date* with domain PE_*Performance*, DATETIME as range, and an extension over INTEGER × DATETIME, see Figure 4, corresponds to the relation schema PE_DAT_*Performance_Date*(PE_ID, PE_DAT_*Performance_Date*).

PE_DAT_*Performance_Date*

PE_ID *(PK, FK)*	PE_DAT_*Performance_Date*
#911	1994-12-10 20:00
#912	1994-12-15 21:30
#913	1994-12-22 19:30
INTEGER	DATETIME

Figure 4: Example rows from the static attribute PE_DAT_*Performance_Date*.

Historized Attributes

A historized attribute B_H with domain A, range $\mathbb{D}$ and time range $\mathbb{T}$ is mapped to a relation schema $B(I, V, P)$, where $I = N_{\mathbb{I}A}(A)$, $V = N_{\mathbb{D}}(B)$ and $P = N_{\mathbb{T}}(B)$. The domain of I is $\mathbb{I}$, the domain of V is $\mathbb{D}$ and the domain of P is $\mathbb{T}$. The columns I and P together constitute a primary key of the relation schema B. The column I is a foreign key with respect to the relation schema $A(S)$.

AC_NAM_*Actor_Name*

AC_ID *(PK, FK)*	AC_NAM_*Actor_Name*	AC_NAM_ValidFrom *(PK)*
#4711	'Arfle B. Gloop'	1972-08-20
#4711	'Voider Foobar'	1999-09-19
#4712	'Nathalie Roba'	1989-09-21
INTEGER	STRING	DATE

Figure 5: Example rows from the historized attribute AC_NAM_*Actor_Name*.

For example the historized attribute AC_NAM_*Actor_Name* with AC_*Actor* as domain, DATE as time range, and an extension over INTEGER × STRING × DATE, see Figure 5, corresponds to the relation schema AC_NAM_*Actor_Name*(AC_ID, AC_NAM_*Actor_Name*, AC_NAM_ValidFrom).

Knotted Static Attributes

A knotted static attribute B_{KS} with domain A and range K is mapped to a relation schema $B(I_1, I_2)$, where $I_1 = N_{\mathbb{I}A}(A)$ and $I_2 = N_{\mathbb{I}K}(K)$. The domain of I_1 is $\mathbb{I}$ and the domain of I_2 is $\mathbb{I}$. The column I_1 is a primary key of B and a foreign key with respect to the relation schema $A(S)$. The column I_2 is a foreign key with respect to the relation schema $K(I, V)$.

AC_GEN_*Actor_Gender*

AC_ID *(PK, FK)*	GEN_ID *(FK)*
#4711	#0
#4712	#1
#4713	#1
INTEGER	BIT

Figure 6: Example rows from the knotted static attribute AC_GEN_*Actor_Gender*.

For example the knotted static attribute AC_GEN_*Actor_Gender* with AC_*Actor* as domain, GEN_*Gender* as range, and an extension over INTEGER × BIT, see Figure 6 corresponds to the relation schema AC_GEN_*Actor_Gender*(AC_ID, GEN_ID).

Knotted Historized Attributes

A knotted historized attribute B_{KH} with domain A, range K, and time range $\mathbb{T}$ is mapped to a relation schema $B(I_1, I_2, P)$, where $I_1 = N_{\mathbb{I}A}(A)$, $I_2 = N_{\mathbb{I}K}(K)$ and $P = N_{\mathbb{T}}(B)$. The domain of I_1 is $\mathbb{I}$, the domain of I_2 is $\mathbb{I}$ and the domain of P is $\mathbb{T}$. The columns I_1 and P together constitute a primary key of the relation schema B. The column I_1 is a foreign key with respect to the relation schema $A(S)$. The column I_2 is a foreign key with respect to the relation schema $K(I, V)$.

AC_PLV_*Actor_ProfessionalLevel*

AC_ID *(PK, FK)*	PLV_ID *(FK)*	AC_PLV_ValidFrom *(PK)*
#4711	#4	1999-04-21
#4711	#5	2003-08-21
#4712	#3	1999-04-21
INTEGER	TINYINT	DATE

Figure 7: Example rows from the knotted historized attribute AC_PLV_*Actor_ProfessionalLevel*.

For example the knotted historized attribute AC_PLV_*Actor_ProfessionalLevel* with AC_*Actor* as domain, PLV_*ProfessionalLevel* as range, DATE as time range, and an extension over INTEGER × TINYINT × DATE, see Figure 7, corresponds to the relation schema AC_PLV_*Actor_ProfessionalLevel*(AC_ID, PLV_ID, AC_PLV_ValidFrom).

Static Ties

A static tie $T_S = \{R_1, \ldots, R_n\}$, $n \geq 2$, is mapped to a relation schema $T(M_1, \ldots, M_n)$ using an order-preserving map, such that R_i is mapped to M_i given a total order on T_S. Every column M_i is a foreign key with respect to the relation schema $A_i(S)$, where A_i is the type of the anchor role R_i, $M_i = N_{\mathbb{I}A}(A_i)$, and the domain of M_i is the type of R_i. The primary key of the relation schema T is I where I is the identifier of the static tie T_S.

PE_*in*_AC_*wasCast*

PE_ID_*in (PK, FK)*	AC_ID_*wasCast (PK, FK)*
#911	#4711
#912	#4711
#913	#4712
INTEGER	INTEGER

Figure 8: Example rows from the static tie PE_*in*_AC_*wasCast*.

For example the static tie PE_*in*_AC_*wasCast* = {*in*, *wasCast*} having the type of *in* as the anchor PE_*Performance*, the type of *wasCast* as the anchor PE_*Performance*, and an extension over INTEGER × INTEGER, see Figure 8, corresponds to the relation schema PE_*in*_AC_*wasCast*(PE_ID_*in*, AC_ID_*wasCast*), with the primary key PE_ID_*in*, AC_ID_*wasCast*.

Historized Ties

A historized tie $T_H = \{R_1, \ldots, R_n, P\}$, $n \geq 2$ is mapped to a relation schema $T(M_1, \ldots, M_n, P)$ using an order-preserving map, such that R_i is mapped to M_i given a total order on T_H. Every column M_i is a foreign key with respect to the relation schema $A_i(S)$, where A_i is the type of the anchor role R_i, $M_i = N_{\mathbb{I}A}(A_i)$, the domain of M_i is the type of R_i, and $P = N_{\mathbb{T}}(B)$ is a column with the domain $\mathbb{T}$. The primary key of the relation schema T is I where I is the identifier of the historized tie T_H.

ST_*atLocation*_PR_*isPlaying*

ST_ID_*atLocation* *(PK, FK)*	PR_ID_*isPlaying* *(FK)*	ST_*atLocation*_PR_*isPlaying*_ValidFrom *(PK)*
#55	#17	2003-12-13
#55	#23	2004-04-01
#56	#17	2003-12-31
INTEGER	INTEGER	DATE

Figure 9: Example rows from the historized tie ST_*atLocation*_PR_*isPlaying*.

For example the historized tie ST_*atLocation*_PR_*isPlaying* = {*atLocation*, *isPlaying*, DATE} having the type of *atLocation* as the anchor ST_*Stage*, the type of *isPlaying* as the anchor PR_*Program*, and an extension over INTEGER × INTEGER × DATE, see Figure 9, corresponds to the relation schema ST_*atLocation*_PR_*isPlaying*(ST_ID_*atLocation*, PR_ID_*isPlaying*, ST_*atLocation*_PR_*isPlaying*_ValidFrom), with the primary key ST_ID_*atLocation*, ST_*atLocation*_PR_*isPlaying*_ValidFrom.

Knotted Static Ties

A knotted static tie $T_{KS} = \{R_1, \ldots, R_n, S_1, \ldots, S_m\}$, $n \geq 2$ and $m \geq 1$, is mapped to a relation schema $T(M_1, \ldots, M_n, N_1, \ldots, N_m)$ using an order-preserving map, such that R_i is mapped to M_i and S_j is mapped to N_j given a total order on T_{KS}. Every column M_i is a foreign key with respect to the relation schema $A_i(S)$, where A_i is the type of the anchor role R_i, $M_i = N_{\mathbb{I}A}(A_i)$, and the domain of M_i is the type of R_i. Every column N_j is a foreign key with respect to the relation schema $K_j(I, V)$, where K_j is the type of the knot role S_j, $N_j = N_{\mathbb{I}K}(K_j)$, and the domain of N_j is the type of S_j. The primary key of the relation schema T is I where I is the identifier of the knotted static tie T_{KS}.

AC_*parent*_AC_*child*_PAT_*having*

AC_ID_*parent* *(PK, FK)*	AC_ID_*child* *(PK, FK)*	PAT_ID_*having* *(PK, FK)*
#4711	#4713	#1
#4712	#4713	#0
#4712	#4714	#0
INTEGER	INTEGER	BIT

Figure 10: Example rows from the knotted static tie AC_*parent*_AC_*child*_PAT_*having*.

For example the knotted static tie AC_*parent*_AC_*child*_PAT_*having* = {*parent*, *child*, *having*} having the type of *parent* and *child* as the anchor AC_*Actor*, the type of *having* as the knot PAT_*ParentalType*, and an extension over INTEGER × INTEGER × BIT, see Figure 10, corresponds to the relation schema AC_*parent*_AC_*child*_PAT_*having*(AC_ID_*parent*, AC_ID_*child*, PAT_ID_*having*), with the primary key AC_ID_*parent*, AC_ID_*child*, PAT_ID_*having*.

Knotted Historized Ties

A knotted historized tie $T_{KH} = \{R_1, \ldots, R_n, S_1, \ldots, S_m, P\}$, $n \geq 2$ and $m \geq 1$, is mapped to a relation schema $T(M_1, \ldots, M_n, N_1, \ldots, N_m, P)$ using an order-preserving map, such that R_i is mapped to M_i and S_j is mapped to N_j given a total order on T_{KH}. Every column M_i is a foreign key with respect to the relation schema $A_i(S)$, where A_i is the type of the anchor role R_i, $M_i = N_{\mathbb{I}A}(A_i)$, and the domain of M_i is the type of R_i. Every column N_j is a foreign key with respect to the relation schema $K_j(I, V)$, where K_j is the type of the knot role S_j, $N_j = N_{\mathbb{I}K}(K_j)$, and the domain of N_j is the type of S_j. The column $P = N_{\mathbb{T}}(B)$ has the domain $\mathbb{T}$. The primary key of the relation schema T is I where I is the identifier of the knotted historized tie T_{KH}.

AC_*part*_PR_*in*_RAT_*got*

AC_ID_*part* *(PK, FK)*	PR_ID_*in* *(PK, FK)*	RAT_ID_*got* *(FK)*	AC_*part*_PR_*in*_RAT_*got*_ValidFrom *(PK)*
#4711	#17	#8	2005-03-12
#4711	#17	#10	2008-08-29
#4712	#17	#10	2006-10-17
INTEGER	INTEGER	TINYINT	DATE

Figure 11: Example rows from the knotted historized tie AC_*part*_PR_*in*_RAT_*got*.

For example the knotted historized tie AC_*part*_PR_*in*_RAT_*got* = {*part*, *in*, *got*, DATE} having the type of *part* as the anchor AC_*Actor*, the type of *in* as the anchor PR_*Program*, the type of *got* as the knot RAT_*Rating*, and an extension over INTEGER × INTEGER × TINYINT × DATE, see Figure 11, corresponds to the relation schema AC_*part*_PR_*in*_RAT_*got*(AC_ID_*part*, PR_ID_*in*, RAT_ID_*got*, AC_*part*_PR_*in*_RAT_*got*_ValidFrom), with the primary key AC_ID_*part*, PR_ID_*in*, AC_*part*_PR_*in*_RAT_*got*_ValidFrom.

Example Schema

The relational database schema resulting from the Anchor model in Figure 1 is shown in Figure 12. It contains anchor, knot, attribute, and tie schemas in that order, where primary keys are marked with asterisks.

Anchor Schema	
AC_Actor	(AC_ID*)
PR_Program	(PR_ID*)
ST_Stage	(ST_ID*)
PE_Performance	(PE_ID*)
Knot Schema	
GEN_Gender	(GEN_ID*, GEN_Gender)
PLV_ProfessionalLevel	(PLV_ID*, PLV_ProfessionalLevel)
RAT_Rating	(RAT_ID*, RAT_Rating)
PAT_ParentalType	(PAT_ID*, PAT_ParentalType)
Attribute Schema	
AC_GEN_Actor_Gender	(AC_ID*, GEN_ID)
AC_PLV_Actor_ProfessionalLevel	(AC_ID*, PLV_ID, AC_PLV_ValidFrom*)
AC_NAM_Actor_Name	(AC_ID*, AC_NAM_Actor_Name, AC_PLV_ValidFrom*)
PR_NAM_Program_Name	(PR_ID*, PR_NAM_Program_Name)
ST_LOC_Stage_Location	(ST_ID*, ST_LOC_Stage_Location)
ST_NAM_Stage_Name	(ST_ID*, ST_NAM_Stage_Name, ST_NAM_ValidFrom*)
PE_DAT_Performance_Date	(PE_ID*, PE_DAT_Performance_Date)
PE_REV_Performance_Revenue	(PE_ID*, PE_REV_Performance_Revenue)
PE_AUD_Performance_Audience	(PE_ID*, PE_AUD_Performance_Audience)
Tie Schema	
AC_parent_AC_child_PAT_having	(AC_ID_parent*, AC_ID_child*, PAT_ID_having*)
AC_part_PR_in_RAT_got	(AC_ID_part*, PR_ID_in*, RAT_ID_got, AC_part_PR_in_RAT_got_ValidFrom*)
ST_atLocation_PR_isPlaying	(ST_ID_atLocation*, PR_ID_isPlaying, ST_atLocation_PR_isPlaying_ValidFrom*)
PE_in_AC_wasCast	(PE_ID_in*, AC_ID_wasCast*)
PE_at_PR_wasPlayed	(PE_ID_at*, PR_ID_wasPlayed)
PE_wasHeld_ST_atLocation	(PE_ID_wasHeld*, ST_ID_atLocation)

Figure 12: The relational schema for the anchor schema in Figure 1, with primary keys marked by asterisks.