SAP-FOX -> NIS Integration Requirements

This document describes the requirements for integrating SAP-FOX to NeXT Improve System. In particular it tries to answer to the following questions:

1. What are SAP-FOX data required by NIS ?
2. When are SAP-FOX data required by NIS ?
3. How could NIS acquire data from SAP-FOX ?
4. What is already implemented into Berbentina plant ?

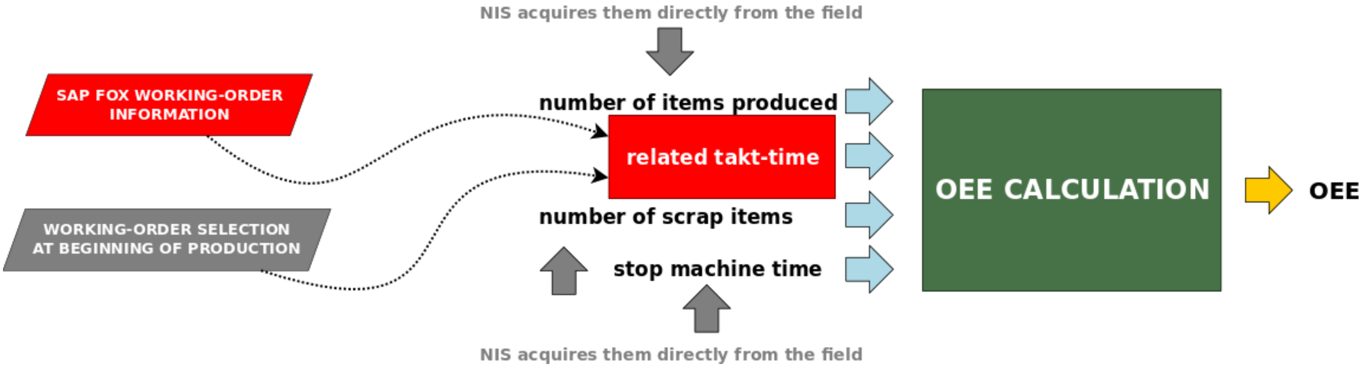
Definitions

Word	**Definition**
SAP-FOX	SAP-FOX ERP System
NIS	NeXT Improve System for OEE Monitoring
ERP-User	Plant's staff that uses **SAP-FOX**
SAP-FOX integration procedure	Procedure implemented into SAP-FOX by SAP-FOX Developers for integrating NIS
WO / **work-order**	Working-Order referred to one specific kind of product and related quantity to be produced
DBMS	Database Management System
NIS DB	**NIS** database that is a relational database using Postgres **DBMS**
SAP-FOX DB	SAP-FOX database
SQL	Structured query language
ruek	identifier used at Berbentina Plant to reach working-order from HMI-barcode-reader into production line; if **SAP-FOX** doesn't use this concept, it will be not considered
RFC	SAP-FOX Remote function call
Web-services	Web-services interface for calling RFC
OEE	Overall Equipment Effectiveness
KPI	Key Performance Indicator

NIS requirements

NIS's target is to calculate OEE KPI for each production line or each machine that should be monitored by it. OEE calculation requires the following data gathered from the field. Those information are referred to a specific available time and to a specific production line / machine:

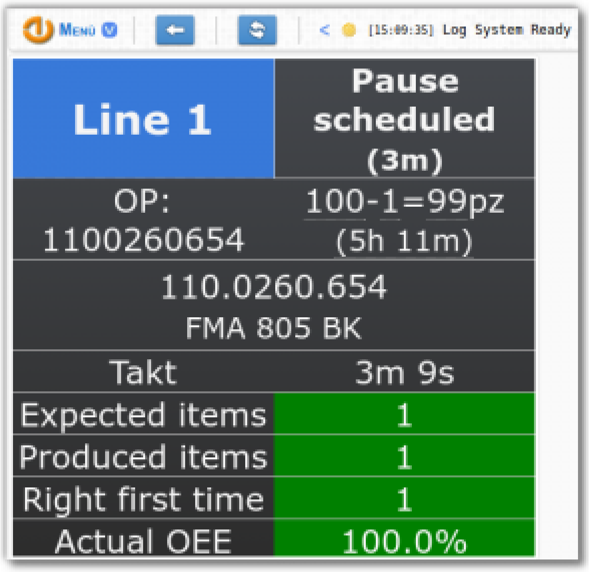
1. number of items produced and related takt-time
2. number of scrap items
3. stop machine time


NIS acquires them directly from the field
SAP FOX WORKING-ORDER INFORMATION
WORKING-ORDER SELECTION AT BEGINNING OF PRODUCTION
number of items produced
related takt-time
number of scrap items
stop machine time
NIS acquires them directly from the field
OEE CALCULATION
OEE

Number of items produced, number of scrap, stop machine should be acquired by NIS directly from the field through NIS's devices. In this way NIS calculates and stores OEE at specific time horizon (each hour, each shift, each day).

Furthermore NIS shows some real-time production information for each production line; following information shown by NIS at real-time :

- WO identifier
- product identifier
- product description
- number of items to be produced (to fulfill WO)



< [15:09:35] Log System Ready

Line 1	Pause scheduled (3m)
OP: 1100260654	100-1=99pz (5h 11m)
110.0260.654 FMA 805 BK	
Takt	3m 9s
Expected items	1
Produced items	1
Right first time	1
Actual OEE	100.0%

Production information are related to WO; moreover takt-time is related to WO applied to a specific production line.

Production information and takt time should be available at any time.

Conceptual Schema

Based on scenario described at previous paragraph, NIS requires working-order's information below:

- **WO identifier**: unique identifier of the working-order
- **plant identifier**: unique identifier of Franke's plant related to WO
- **product identifier and description**: code and description of the product referred by the WO
- **quantity to be produced**: number of items to be produced

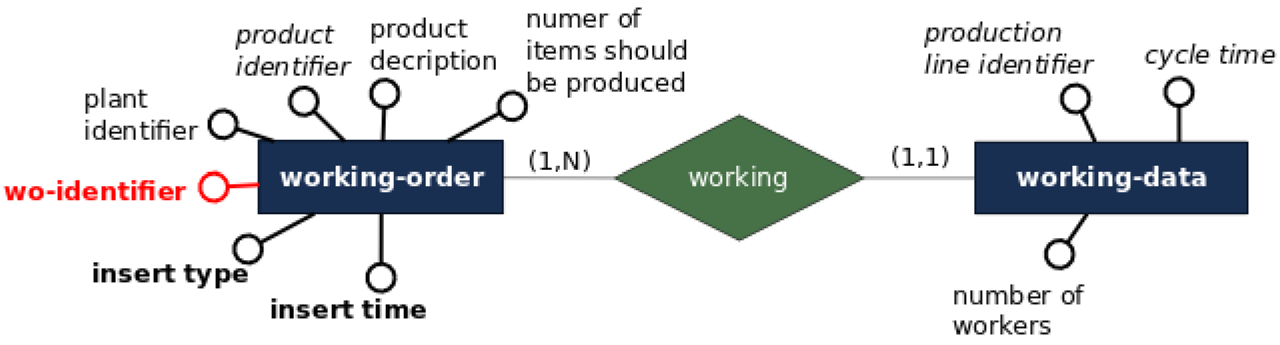
Each WO refers also to following production information for each production line / machine where WO could be produced:

- **production line**: code of the production line (assembly line / machine) interested by WO
- **cycle time**: the period in seconds, required to complete one cycle of production (production of one item); it corresponds exactly to takt time in case of mechanical line
- **number of workers**: number of workers assigned to WO (significant only for assembly line for calculating takt time through cycle time)

It's necessary to manage some flags that permit the identification of working-order's information status; following:

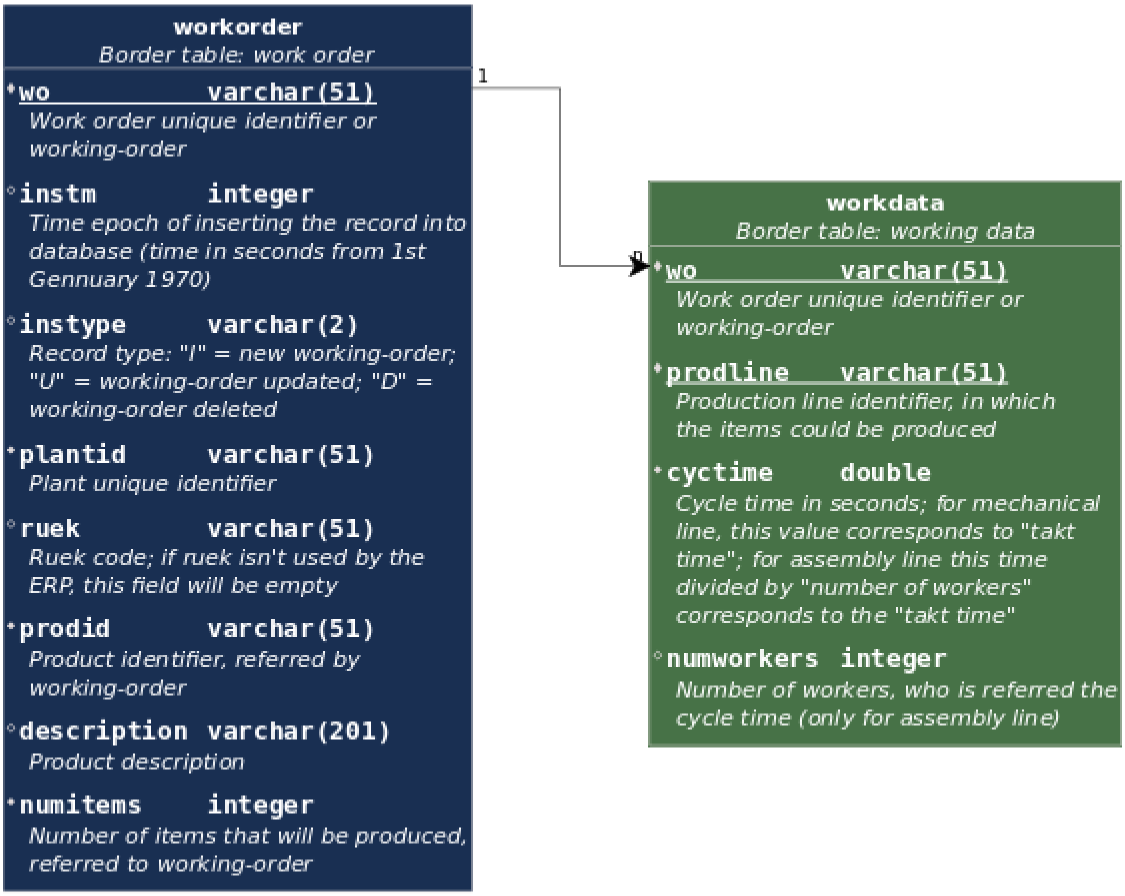
- **time status (insert time)**: time in which the information is generated
- **information status (insert type)**: status of the information, distinguishing as following:
  - *inserted information*: information to be inserted at first time
  - *updated information*: information to be updated
  - *deleted information*: information to be deleted

Following a conceptual schema:



Logical Schema

NIS integration to SAP-FOX should be applied using some shared tables; only two shared tables are sufficient for implementing integration. Following logical schema genereted from previous conceptual schema



workorder
Border table: work order

*wo varchar(51)
Work order unique identifier or working-order

°instm integer
Time epoch of inserting the record into database (time in seconds from 1st Gennuary 1970)

°instype varchar(2)
Record type: "I" = new working-order; "U" = working-order updated; "D" = working-order deleted

*plantid varchar(51)
Plant unique identifier

°ruek varchar(51)
Ruek code; if ruek isn't used by the ERP, this field will be empty

*prodid varchar(51)
Product identifier, referred by working-order

°description varchar(201)
Product description

*numitems integer
Number of items that will be produced, referred to working-order

workdata
Border table: working data

*wo varchar(51)
Work order unique identifier or working-order

*prodline varchar(51)
Production line identifier, in which the items could be produced

*cyctime double
Cycle time in seconds; for mechanical line, this value corresponds to "takt time"; for assembly line this time divided by "number of workers" corresponds to the "takt time"

°numworkers integer
Number of workers, who is referred the cycle time (only for assembly line)

The name of first shared table is "workorder", following fields list:

Field Name	Field Type	Field Description	Primary key	Nullable
wo	varchar(51)	Unique identifier of working-order	Yes	No
instm	integer	Time epoch of inserting the record into database (time in seconds from 1st Gennuary 1970)	No	No
instype	varchar(2)	Record type: "I" = new working-order; "U" = working-order updated; "D" = working-order deleted	No	No
plantid	varchar(51)	Plant unique identifier	No	No
ruek	varchar(51)	Ruek code; if ruek isn't used by the ERP, this field will be empty	No	Yes
prodid	varchar(51)	Product identifier, referred by working-order	No	No
description	varchar(201)	Product description	No	Yes
numitems	integer	Number of items that will be produced, referred to working-order	No	No

The primary key is defined on the following fields:

- **<wo>**

It implies that a specific WO is related to a specific row of the table workorder; as field **wo** is primary key only last WO's version should be present in the workorder table. Furthermore when a WO is updated/deleted by ERP User, it's necessary to update **instm** and **instype** according to their semantic (described in the table of fields).

The name of second shared table is "workdata", following fields list:

Field Name	Field Type	Field Description	Primary key	Nullable
wo	varchar(51)	Unique identifier of working-order	Yes	No

Field Name	Field Type	Field Description	Primary key	Nullable
prodline	varchar(51)	Production line identifier, in which the items could be produced	Yes	No
cycletime	double	Cycle time in seconds; for mechanical line, this value corresponds to "takt time"; for assembly line this time divided by "number of workers" corresponds to the "takt time"	No	No
numworkers	integer	Number of workers, who is referred the cycle time (only for assembly line)	No	Yes

The primary key is defined on the following fields:

- **<wo, prodline>**

A foreign key is defined on field **wo** (it refers to field wo of table workorder). The relation between table **workorder** and table **workdata** is 1 to N respectively (as we can see into conceptual schema and also into logical schema). Therefore one working-order stored into **workorder** table corresponds to some rows into **workdata** table, one row for each production line in which working-order could be produced. This relation permits to deduce takt-time for each WO for each production line where WO could be produced.

Following an example of create table query for creating workorder/workdata table using standard SQL language

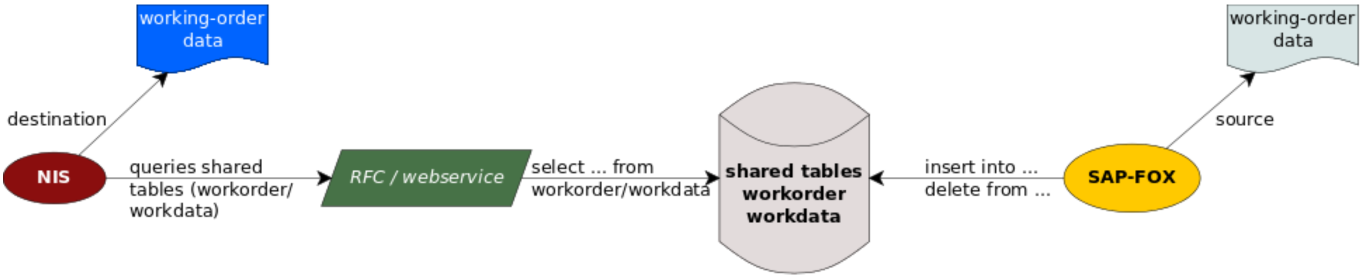
```sql
--working-order table and related constraint
CREATE TABLE workorder (
    wo VARCHAR(51) NOT NULL,
    instm INTEGER NOT NULL,
    instype VARCHAR(2) NOT NULL,
    plantid VARCHAR(51) NOT NULL,
    ruek VARCHAR(51),
    prodid VARCHAR(51) NOT NULL,
    description VARCHAR(201),
    numitems INTEGER NOT NULL,
    prodline VARCHAR(51) NOT NULL,
    cycletime FLOAT NOT NULL,
    numworkers INTEGER
);
ALTER TABLE workorder
    ADD CONSTRAINT ix_workorder_key PRIMARY KEY (wo, instm, instype);
--working-data table and related constraint
CREATE TABLE workdata (
    wo CHARACTER VARYING(51) NOT NULL,
    prodline CHARACTER VARYING(51) NOT NULL,
    cycletime DOUBLE PRECISION NOT NULL,
    numworkers INTEGER
);
ALTER TABLE workdata
    ADD CONSTRAINT ix_workdata_key PRIMARY KEY (wo, prodline);
ALTER TABLE workdata
    ADD CONSTRAINT ix_workdata_forkey FOREIGN KEY (wo) REFERENCES workorder(wo);
```

The shared tables workorder and workdata should be placed into SAP-FOX database; NIS should query workorder/workdata through RFC/web-service call to SAP-FOX. While NIS queries shared tables then SAP-FOX returns all rows required and removes them from shared tables (or marks them as NIS received).

Integration Criterions

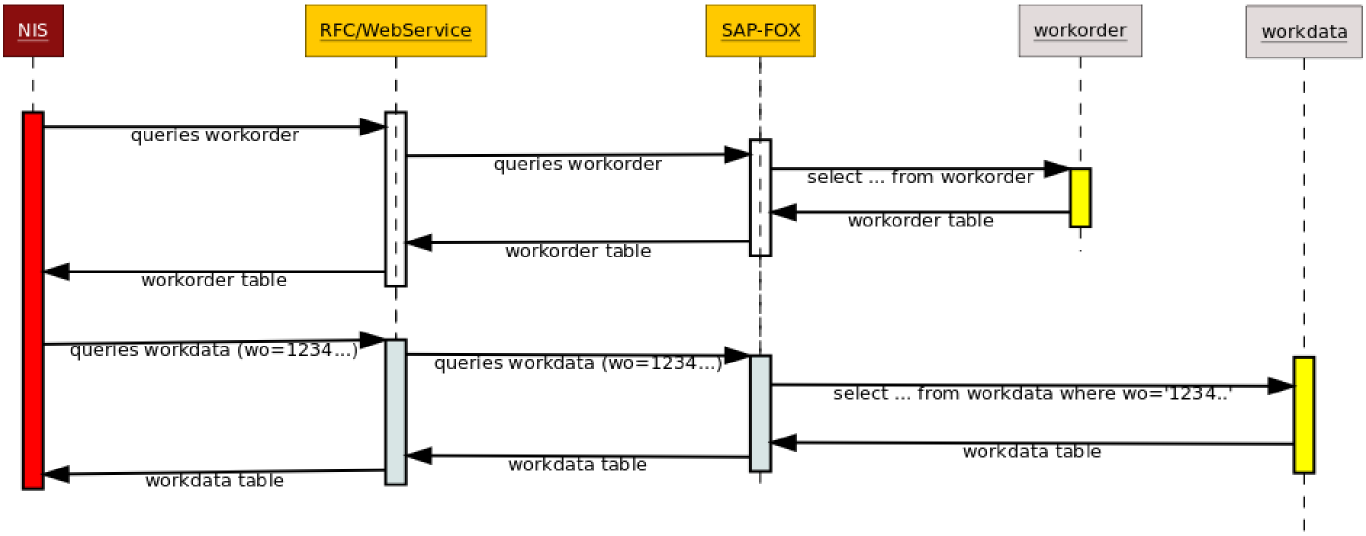
The integration between NIS and SAP-FOX could be implemented using RFC (or web-services) defined into SAP-FOX.

NIS should be able to query SAP-FOX using RFC/web-service for acquiring required information and then it should insert them into NIS DB. RFC/Webservice should be implemented into SAP-FOX by SAP-FOX developers (SAP-FOX integration procedures). Following a general flow-chart


working-order data
destination
NIS
queries shared tables (workorder/ workdata)
RFC / webservice
select ... from workorder/workdata
shared tables workorder workdata
insert into ... delete from ...
SAP-FOX
source
working-order data

Working-order data are the source of information; first they are available into SAP-FOX database and are accessible by SAP-FOX System, but not by NIS. So SAP-FOX inserts information required by NIS into working-order shared tables; shared tables are provided from SAP-FOX DB.

Periodically (for example once at hour) NIS queries shared tables using RFC/web-service. Sequence diagram below describes a possible time-line for interacting:

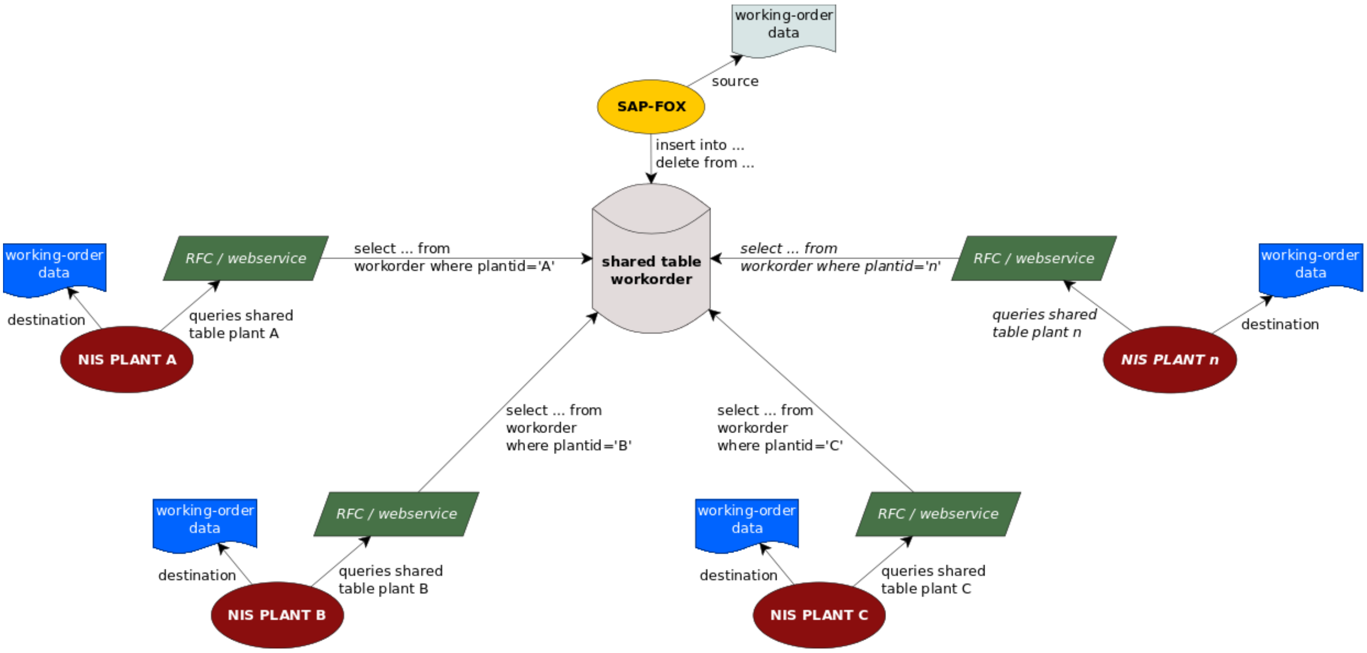


After SAP-FOX inserts data into shared tables, NIS detects and saves them into destination table (operative table). While NIS received data, SAP-FOX could delete/mark them into shared tables (workorder); in this way at next NIS's interrogation deleted/marked data should not returned.

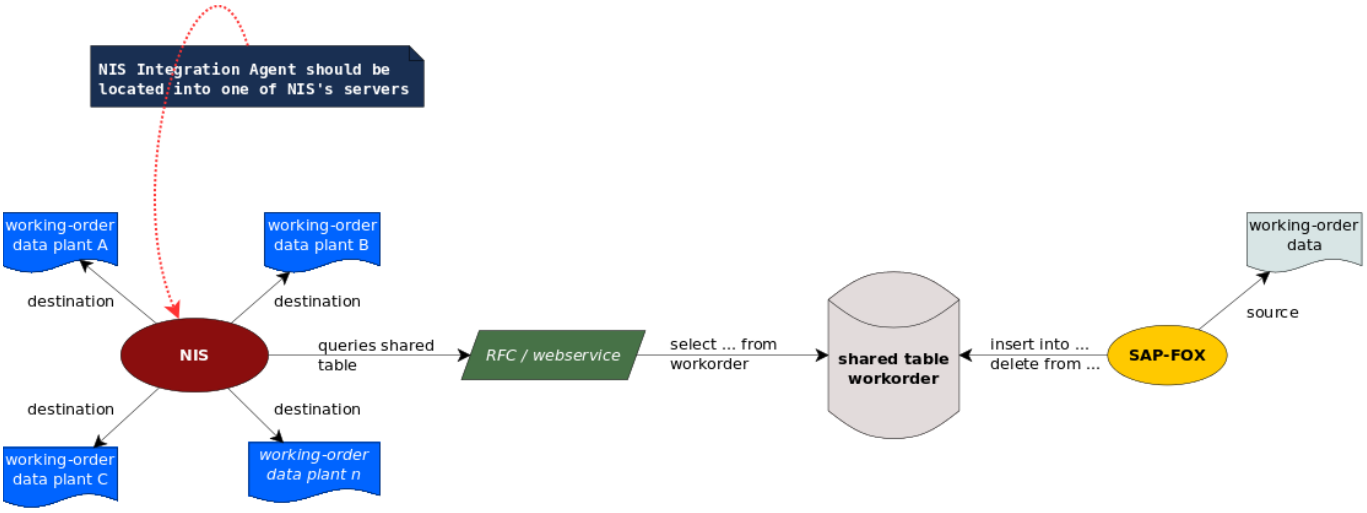
It's sufficient one pair of tables (workorder/workdata) for all managed plants; at this point we could define two integration contexts that could be implemented as following:

1. **NIS queries shared tables plant by plant** using an RFC/webservice parameter that specifies plant identifier and RFC/webservice returns only specific plant's WOs
2. **NIS queries shared tables once for all plants** (without specifies plant identifier) and RFC/webservice returns WOs for every plants

NIS queries workorder plant by plant



NIS queries workorder once

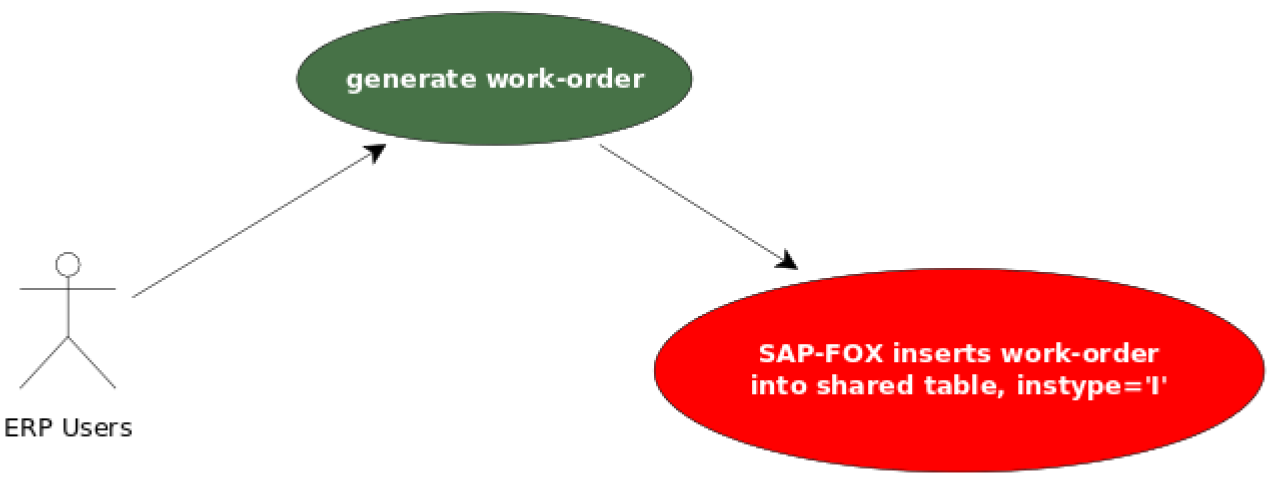

NIS Integration Agent should be located into one of NIS's servers
working-order data plant A
working-order data plant B
destination
destination
NIS
queries shared table
RFC / webservice
select ... from workorder
shared table workorder
insert into ...
delete from ...
SAP-FOX
source
working-order data
destination
destination
working-order data plant C
working-order data plant n

Integration Scenarios

Working-order data may be generated, modified, deleted by ERP-User; so from this point of view NIS ↔ SAP-FOX integration should manage following scenarios:

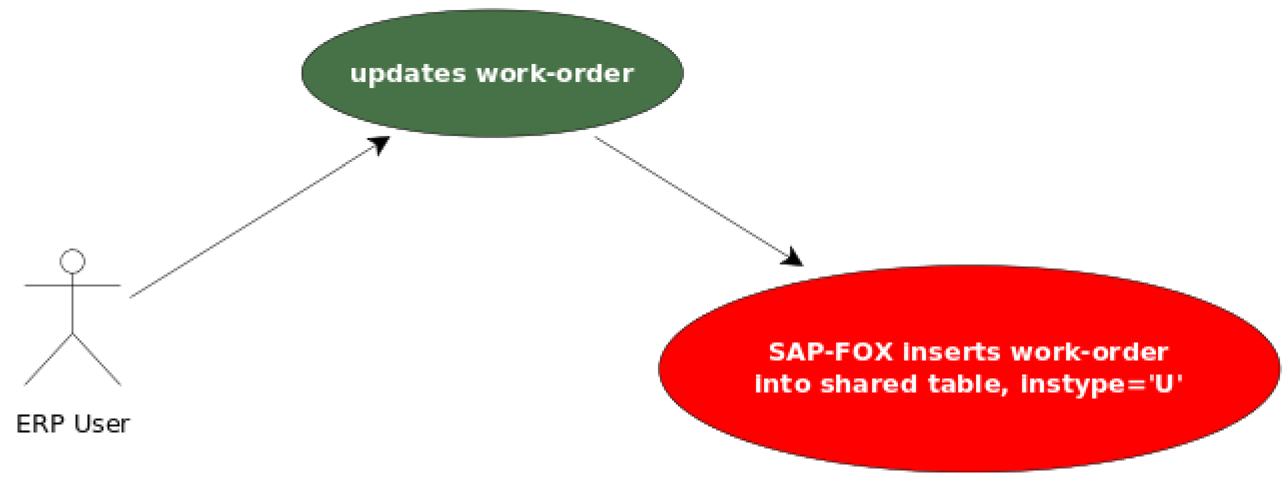
1. ERP-User inserts a new working-order into SAP-FOX
2. ERP-User updates an existing working-order into SAP-FOX
3. ERP-User deletes an existing working-order into SAP-FOX

USER inserts a new WO into ERP



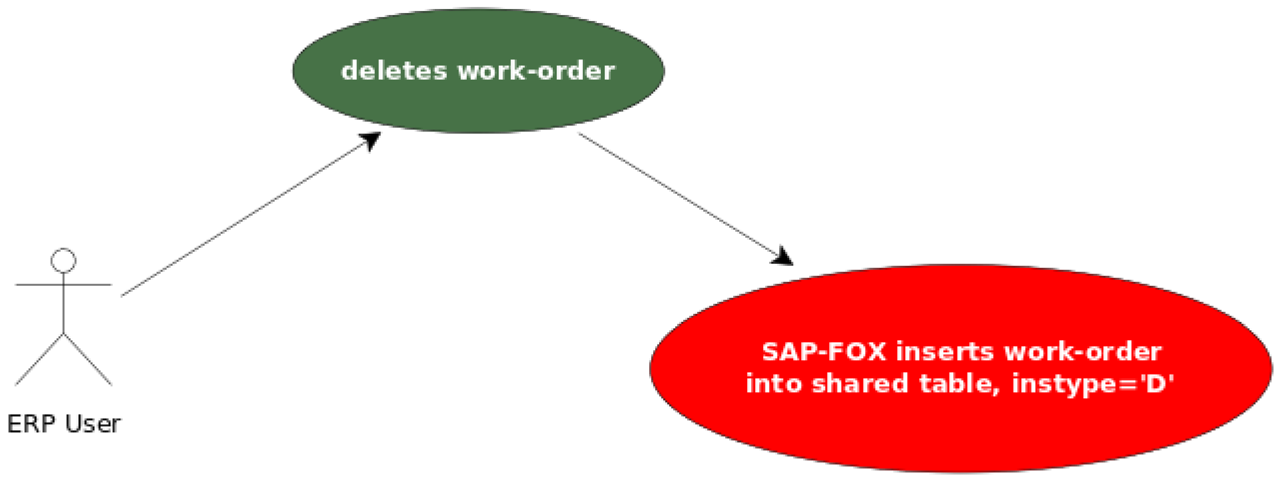
If an **ERP-User** inserts a new working-order into SAP-FOX, then SAP-FOX should insert data required into workorder table and into workdata table; in this case the field **instm** should be set to the time of inserting record into shared table, while **instype** should be set to **I** ([I]nsert).

USER updates an existing WO into ERP



If an **ERP-User** updates an existing working-order into SAP-FOX, then SAP-FOX should delete→insert data required into workorder table and into workdata table; in this case the field **instm** should be set to the time of inserting record into shared table, while **instype** should be set to **U** ([U]pdate).

USER deletes an existing WO into ERP



If an **ERP-User** deletes an existing working-order into SAP-FOX, then SAP-FOX should update data required into workorder table (nothing is required into workdata table); in this case the field **instm** should be set to the time of updating record into shared table, while **instype** should be set to **D** ([D]elete).

Integration Examples

We show an example in which ERP-User, from plant identified by FFI-code, executes following steps:

1. first creates a new working-order identified by 44621852
2. then updates working-order 44621852 for changing number of items
3. finally deletes working-order 44621852

Naturally updating and deleting should be optionally operations and ERP-User should execute them based on his needs.

We begin considering that ERP-User inserts into SAP-FOX the following working-order

WO	PRODUCT ID	QTY(UNIT)
44621852	110.0017.976	20

This working-order could be produced on production line FAFL1 and FAFL2, with the following information

LINE	CYC TIME(sec)	NUM WORKERS
FAFL1	900	5
FAFL2	900	4

SAP-FOX integration procedure should insert data into related fields of the tables workorder / workdata; following relations between data in table above and fields in workorder / workdata tables

workorder		
Field Name	**Data**	**Annotation**
wo	44621852	
instime	1480712220	Corresponds to 2016-12-02 21:57
instype	I	

workorder		
Field Name	**Data**	**Annotation**
plantid	FFI	
ruek	null	
prodid	110.0017.976	
description	COOKER HOOD XXXX	ERP extracts product description from internal table
numitems	20	

workdata			
Num row	**Field Name**	**Data**	**Annotation**
1	wo	44621852	
1	prodline	FAFL1	Improve should know this code
1	cycletime	900	
1	numworkers	5	
2	wo	44621852	
2	prodline	FAFL2	Improve should know this code
2	cycletime	900	
2	numworkers	4	

Following query in SQL language that SAP-FOX should execute on shared table

```sql
INSERT INTO workorder (wo, instm, instype, plantid, prodid, description, numitems)
VALUES ('44621852', 1480712220, 'I', 'FFI', '110.0017.976', 'COOKER HOOD XXXX',
20);
INSERT INTO workdata (wo, prodline, cycletime, numworkers)
VALUES ('44621852', 'FAFL1', 900, 5);
INSERT INTO workdata (wo, prodline, cycletime, numworkers)
VALUES ('44621852', 'FAFL2', 900, 4);
```

For reasons that we don't know, ERP-User updates the number of items from 20 to 30 in the same working-order; after that SAP-FOX updates working-order information as following:

WO	PRODUCT ID	QTY(UNIT)
44621852	110.0017.976	***30***

SAP-FOX integration procedure should insert into shared tables the change occurred. To do this SAP-FOX inserts into table workorder following data:

Field Name	Data	Annotation
wo	44621852	
instime	***1480718520***	***Corresponds to 2016-12-02 23:42***
instype	***M***	
plantid	FFI	
ruek	null	
prodid	110.0017.976	
description	COOKER HOOD XXXX	ERP extracts product description from internal table
numitems	***30***	

The query in SQL language that SAP-FOX should execute on shared table is defined below (we assumed SAP-FOX deletes and inserts again all data):

```sql
DELETE FROM workorder WHERE wo='44621852';
DELETE FROM workdata WHERE wo='44621852';
INSERT INTO workorder (wo, instm, instype, plantid, prodid, description, prodline,
numitems, cycletime, numworkers)
VALUES ('44621852', 1480718520, 'M', 'FFI', '110.0017.976', 'COOKER HOOD XXXX',
'FAFL1', 30, 900, 5);
INSERT INTO workdata (wo, prodline, cycletime, numworkers)
VALUES ('44621852', 'FAFL1', 900, 5);
INSERT INTO workdata (wo, prodline, cycletime, numworkers)
VALUES ('44621852', 'FAFL2', 900, 4);
```

Not enough satisfied ERP-User adjudicates to delete the working-order from SAP-FOX system. SAP-FOX should comunicate this event to NIS, so SAP-FOX inserts into workorder table following data:

Field Name	Data	Annotation
wo	44621852	
instime	***1480719360***	***Corresponds to 2016-12-02 23:56***
instype	***D***	
plantid	FFI	
ruek	null	
prodid	110.0017.976	
description	COOKER HOOD XXXX	ERP extracts product description from internal table
numitems	***30***	

SAP-FOX executes query SQL below on shared table (nothing is required on workdata):

```
DELETE FROM workorder WHERE wo='44621852';
DELETE FROM workdata WHERE wo='44621852';
INSERT INTO workorder (wo, instm, instype, plantid, prodid, description, prodline,
numitems, cycletime, numworkers)
VALUES ('44621852', 1480719360, 'D', 'FFI', '110.0017.976', 'COOKER HOOD XXXX',
'FAFL1', 30, 900, 5);
```

If NIS queries workorder table using RFC/webservice call, after ERP-User updates number of items and before ERP-User deletes WO and (for hypotesis) nothing other is occurred, then it receives following data (SAP-FOX deletes/marks this row as NIS readed):

wo	instime	instype	plantid	ruek	prodid	description	numitems
44621852	1480718520	M	FFI	null	110.0017.976	COOKER HOOD XXXX	30

After who NIS queries workdata requiring WO number 44621852 and receives following data

wo	prodline	cycletime	numworkers
44621852	FAFL1	900	5
44621852	FAFL2	900	4

But if NIS queries workorder table after deleting data then receives following data:

wo	instime	instype	plantid	ruek	prodid	description	numitems
44621852	1480719360	**D**	FFI	null	110.0017.976	COOKER HOOD XXXX	30

So NIS doesn't take into consideration WO 44621852.

Berbentina plant integration

Actual integration NIS ↔ SAP-FOX at Berbentina plant is based on RFC. NIS queries not only WO's information but also items produced. Items produced may be generated by Factory Kit; this software-tools is available only at Berbentina Plant, so every function depends on it doesn't keep in consideration.

We place our focus only on sharing working-order information. At every time NIS could queries SAP-FOX for getting working-order information, using RFC **/FOX/PP_ANDON** (already implemented into SAP-FOX at Berbentina plant).

This RFC receives following import parameters:

Parameter name	Type	Description
I_RUEK	CO_RUEK	Number of operation confirmation
I_HEADER_DATA	BOOLEAN	If true it indicates that it's required working-order header data
I_OPERATIONS_DATA	BOOLEAN	If true it indicates that it's required working-order operation data
I_SERIALNO_DATA	Not used	

If I_HEADER_DATA is true then RFC answers with the following table:

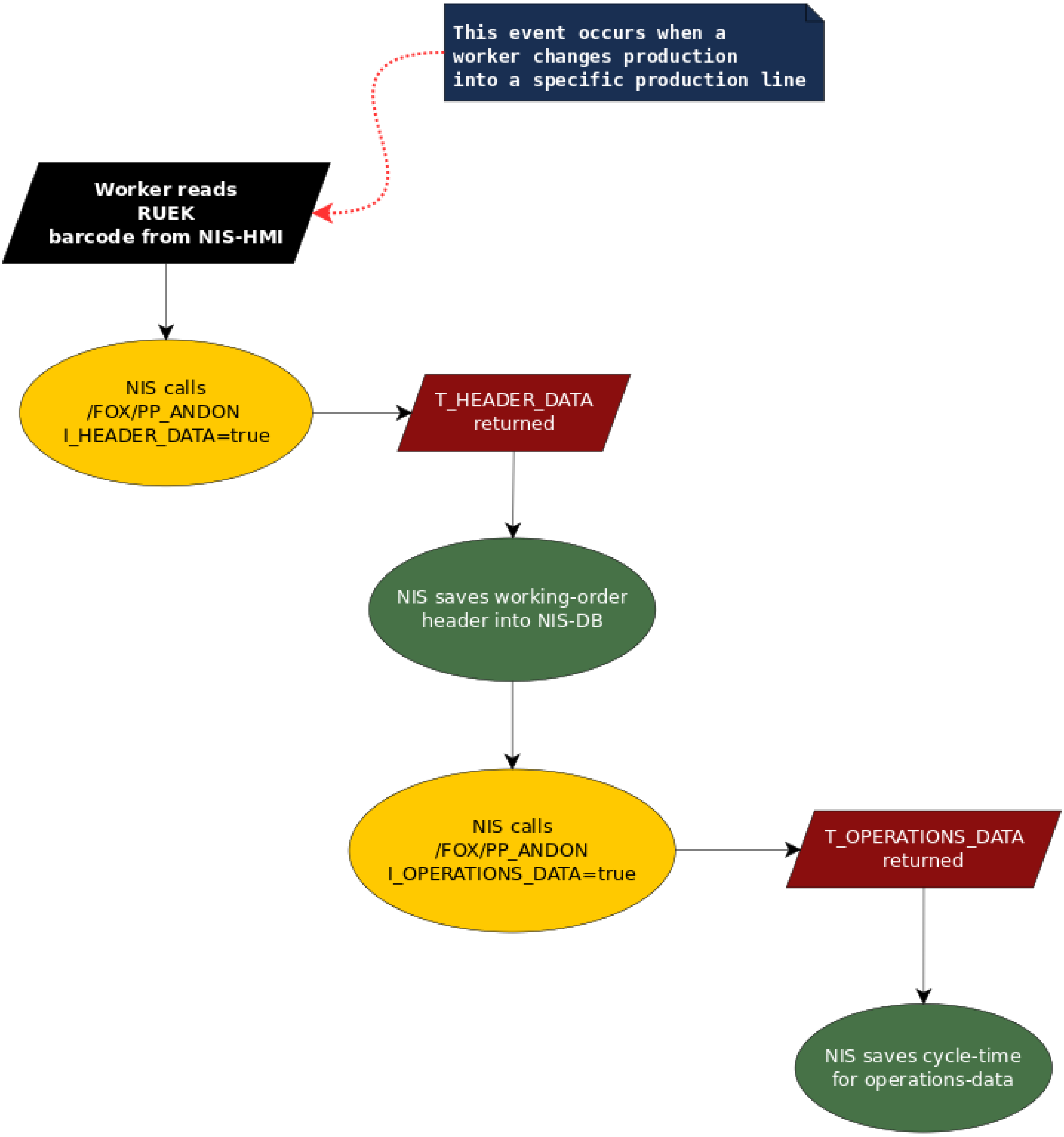
Component	Component type	Field Type	Size	Decimals	Description
DWERK	WERKS_D	CHAR	4	0	Division
AUFNR	AUFNR	CHAR	12	0	Number of order
MATNR	MATNR	CHAR	18	0	Product code
MAKTX	MAKTX	CHAR	40	0	Product description
GSTRP	PM_ORDGSTRP	DATS	0	8	Date of beginning order
GLTRP	CO_GLTRP	DATS	8	0	Date of ending order
GMEIN	MEINS	UNIT	3	0	Unit of measure
GAMNG	GAMNG	QUAN	13	3	Total quantity to be produced
WEMNG	CO_WEMNG	QUAN	13	3	
IGMNG	CO_IGMNG	QUAN	13	3	Confirmed quantity
FK_ANZMA	ANZMS	DEC	5	2	Number of workers
FK_PERNNR	PERNR_D	NUMC	8	0	C.I.D.

If I_OPERATIONS_DATA is true then RFC answers with the following table:

Component	Component type	Field Type	Size	Decimals	Description
WERKS	WERKS_D	CHAR	4	0	Division
AUFNR	AUFNR	CHAR	12	0	Number of order
VORNR	VORNR	CHAR	4	0	Number of operation
STEUS	STEUS	CHAR	4	0	Controlo key
ARBPL	ARBPL	CHAR	8	0	Work center
LTXA1	LTXA1	CHAR	40	0	Operation description
BMSCH	BMSCH	QUAN	13	3	Base quantity
MEINH	MEINH	UNIT	3	0	Unit of measure for visualization
ANZMA	ANZMS	DEC	5	2	Number of workers
VGE01	VGWRTEH	UNIT	3	0	Unit of measure for standard valute
VGW01	VGWRT	QUAN	9	3	Standard value
VGE02	VGWRTEH	UNIT	3	0	Unit of measure for standard valute
VGW02	VGWRT	QUAN	9	3	Standard value
VGE03	VGWRTEH	UNIT	3	0	Unit of measure for standard valute

Component	Component type	Field Type	Size	Decimals	Description
VGW03	VGWRT	QUAN	9	3	Standard value
VGE04	VGWRTEH	UNIT	3	0	Unit of measure for standard valute
VGW04	VGWRT	QUAN	9	3	Standard value
VGE05	VGWRTEH	UNIT	3	0	Unit of measure for standard valute
VGW05	VGWRT	QUAN	9	3	Standard value
VGE06	VGWRTEH	UNIT	3	0	Unit of measure for standard valute
VGW06	VGWRT	QUAN	9	3	Standard value

Flow chart that describes NIS ↔ SAP-FOX integration at Berbentina plant is shown below


This event occurs when a worker changes production into a specific production line
Worker reads RUEK barcode from NIS-HMI
NIS calls /FOX/PP_ANDON I_HEADER_DATA=true
T_HEADER_DATA returned
NIS saves working-order header into NIS-DB
NIS calls /FOX/PP_ANDON I_OPERATIONS_DATA=true
T_OPERATIONS_DATA returned
NIS saves cycle-time for operations-data

T_HEADER_DATA: used fields

- MATNR
- MAKTX
- GSTRP
- GLTRP
- GMEIN
- GAMNG
- WEMNG
- IGMNG
- AUFNR
- FK_ANZMA
- FK_PERNNR
- MATGR

In the following the code that is using the above fields.

```
rv.setValue("codop", rueck);
rv.setValue("codmod", rt.getString("MATNR"));
rv.setValue("descrizio", rt.getString("MAKTX").replaceAll("[^a-zA-Z0-9 _-]+",
"_"));
rv.setValue("din", rt.getString("GSTRP"));
rv.setValue("dfi", rt.getString("GLTRP"));
rv.setValue("um", rt.getString("GMEIN"));
rv.setValue("qtot", rt.getString("GAMNG"));
rv.setValue("q1", rt.getString("WEMNG"));
rv.setValue("qconf", rt.getString("IGMNG"));
rv.setValue("codprog", rt.getString("AUFNR"));
rv.setValue("ndip", rt.getString("FK_ANZMA"));
rv.setValue("codsq", rt.getString("FK_PERNNR"));
rv.setValue("matgr", rt.getString("MATGR"));
```

T_OPERATIONS_DATA: used fields

- VGW04
- ANZMA
- VGW03
- VGE03
- ARBPL
- LTXA1
- BMSCH
- VGW01

In the following the code that is using the above fields.

```
double tc = Double.parseDouble(ro.getString("VGW04"));
if(tc == 0.0){
  double ndi = Double.parseDouble(ro.getString("ANZMA"));
  if(ndi != 0.0){
    tc = Double.parseDouble(ro.getString("VGW03")) / ndi;
  }
}
if(ro.getString("VGE03").equals("S")){
  ///tc gia' in secondi non faccio nulla
}else if(ro.getString("VGE03").equals("MIN")){
  ///converto il tc da minuti a secondi
  tc *= 60;
}else if(ro.getString("VGE03").equals("H") || ro.getString("VGE03").equals("STD")){
  ///converto il tc da ore a secondi
  tc *= 3600;
}
//rv.setValue("codriso", ro.getString("VORNR"));
rv.setValue("codriso", ro.getString("ARBPL"));
rv.setValue("descrizio", ro.getString("LTXA1"));
///unita di misura del tempo (1=minuti)
rv.setValue("unptm", ro.getString("BMSCH"));
///numero di persone dedicate alla linea
rv.setValue("ndiptm", ro.getString("ANZMA"));
///tempo previsto per il setup
rv.setValue("tmsetup", ro.getString("VGW01"));
///tempo ciclo di linea gia' elaborato ovvero e' la cadenza della linea in secondi
rv.setValue("tm", tc);
///tempo dell'intera operazione va diviso per il numero di persone e si ottiene il
tm
rv.setValue("tmoper", ro.getString("VGW03"));
```