

# LOAN OBJECTS

---

## Presentation

# LOAN OBJECTS

---

## Table of contents

- I. From strategy to product
- II. 5 top benefits from building one's IS with *LOAN OBJECTS*
- III. From the 5 top benefits to the levers which brought them
- IV. Focusing on 5 levers
- V. Architecture principles
- VI. Some strong points (from a business point of view)
- VII. Some other strong points (from an IT point of view)

# LOAN OBJECTS

---

## I. From strategy to product

Our strategy :

1. Address the **most solvent** “market segment”
  - “Credit” is the most profitable of all banking business lines
2. At the same time, address a **much larger** “market segment” than most banking software editors do.
  - In most countries, there are much more companies making a business of lending money, than true banks.
  - a “credit management” software would find much more customers than generalist banking softwares
3. Present this segment with the “cheapest ever” IT offer by keeping – from the very start of the application’s lifecycle - to all modern architecture “best practices”

The product :

*LOAN OBJECTS* =

**Detailed** formalized design (to be used as specifications in a development project)

... of a **full international** credit-to-individuals (ie consumer credit + credit cards + mortgage loans + leasing...)

... software

# LOAN OBJECTS

---

## II. 5 **top benefits** from building one's Information System with *LOAN OBJECTS*

1. Meets all needs identified by :

One of Europe's consumer-credit leaders

One of France's mortgage-loan leaders

2. Low development cost

Estimated at less than 1000 men-day by several independent experts

3. Low migration cost

4. Low maintenance cost

5. Low learning cost

*These benefits are justified hereafter →*

# LOAN OBJECTS

---

## III. From top benefits to the levers which brought them (1/2)

### 1. *Extensive coverage of business needs* thanks to :

→ On-the-spot analysis of the needs of the above-mentioned leaders

This applies to **the whole credit lifecycle**, from origination to collection or write-off ...

### 2. *Low development cost* thanks to :

→ Formal language/syntax used\*

Because it is presently the most suited to productive development methods

→ Model optimization\*

Because it keeps down the number of components (“classes” and “methods”)

→ Highest possible detail level\*

Because it prevents errors, and as a consequence brings down the length of tests

### 3. *Low migration cost* thanks to :

→ Model optimization\*

Because it keeps down the number of target classes in the migration

→ Formal language/syntax used\*

Because it makes plain and simple the meaning of target data, so migration rules are easy to set up

→ Adaptability to any organization\*

→ For the French market mainly : our “initial settings” proposal

\* *Focus on these levers on page 7*

# LOAN OBJECTS

---

## III. From the 5 benefits to the levers which brought them (1/2)

### 4. *Low maintenance cost* thanks to :

→ Formal language/syntax used\*

Because it is nowadays the most suited to productive development methods.

→ Maximal use of parameters\*

Because it allows the software to evolve without resorting to true developments

### 5. *Low learning costs* thanks to :

→ Formal language/syntax chosen\*

Because the user's manual may be easily derived from our documents

→ Highest possible detail level\*

Because it allows business rules to be known from anybody in the bank, at the most detailed level

→ Model optimization\*

Which brings about a marked simplicity of the underlying software, most of the operations unfolding along the same lines

\* *Focus on these levers on page 7*

# LOAN OBJECTS

---

## IV. Focusing on 5 among the above levers

### 1. Formal language/syntaxis used

This language is a subset of UML 2.0 with a few add-ons

The whole design is almost **absolutely abbreviation-free**, which makes it quite reader-friendly.

### 2. Highest possible detail level

This detail level allows to foresee **with 100% certainty** all apparent effects of the underlying software

This is why *LOAN OBJECTS* may be used “as is” as specifications in an IT contract.

### 3. Conceptual optimization

It is **impossible**, unless giving up lever #2, to design with fewer words a software with the same functional coverage.

### 4. Maximal use of parameters

Any business rule, as long as it is not necessarily imposed on all banks around the world, may be freely **adapted** by the user-company

### 5. Adaptability to any organization

It is **impossible**, unless leaving the Information System in an incoherent state, to split any of the designed “operations” into more elementary ones.

We pledge ourselves, contractually, to each of these 5 levers

# LOAN OBJECTS

---

## V. Architecture principles (1/4)

We may think any Information System as made up of 4 distinct sub-domains :

### Sub-domain 1 :

Definition : what may also be found in businesses other than credit-to-individuals.

Examples :

- Operator management
- Business process design
- Appointment management
- Mailing

Usually recommended solution for this sub-domain : Off-the-peg software packages

For instance :

- Directory management software
- Business Process management software
- Word processing software

→ *LOAN OBJECTS* does not cover this sub-domain



# LOAN OBJECTS

---

## V. Architecture principles (2/4)

### Sub-domain 2 :

Definition : what is **specific to the credit-to-individual** business, but is necessarily shared by all banks.

Our contribution : Functional design of the sub-domain

### Sub-domain 3 :

Definition : what **should be customized** by each bank

Examples :

- Loan granting rules (scores...)
- How to compute interests as a function of capital, unpaid interests, unpaid insurance fees...
- Should a direct debit be emitted for the account ?

Usually recommended solution for this sub-domain : manage this domain with a “Business Rules Management Systems” or BRMS

Our contribution : is twofold :

- In sub-domain 2, we have spotted all “rules” that must be viewed as “customizable” and their “input” parameters, without filling in their body
- Our document “LOAN OBJECTS / Initial settings” is an example of rules fitting a middle-sized company selling credit in France

# LOAN OBJECTS

---

## V. Architecture principles (3/4)

### Sub-domain 4 :

Definition : processings that read large quantities of data without updating them

Examples :

- Accounting
- Basel 2 reporting
- Scorecards

Usually recommended solution for this sub-domain : “Data Warehouse”

Our contribution : Our design includes all historical data which belong only to this sub-domain

- The link between “historical” and “primary” data is made plain and simple
- This includes all balances and entries useful for accounting

# LOAN OBJECTS

---

## V. Architecture principles : Conclusion

In order to get a full applicative solution around the software designed in *LOAN OBJECTS*, you still have :

1. To link it to 2 or 3 non-"business specific" softwares, at least :
  - a BPM software
  - a word processing software
2. To link it to a "Business Rule Management System" Software  
For instance : ILOG Rules, de ILOG/IBM
3. To plug the reporting applications (in particular, general accounting) to the physical implementation of the "Data Warehouse" designed in *LOAN OBJECTS*

# LOAN OBJECTS

---

## VI. Some strong points – business point of view (1/3)

- Suited to any national market
- Strong help to selling
  - As soon as a client/contract is selected, a table of all possible/recommended offers is displayed
- Almost any service may be charged to the customer or to a third part
- No more customer “doubles” thanks to :
  - Putting aside customers you have not yet had time to compare with others
  - Powerful operations “merging” efficiently customer doubles
- Account balances are always exact *in real time*
- In case of error, all or part of the billing (interests, fees...) may be recalculated accurately from any chosen date onwards
- “Credit packages” management
- (personal) Guarantees management
- Debt consolidation management
- “Loyalty programs” management
  - (Useful since retailing groups often entrust them to their financial subsidiary)

# LOAN OBJECTS

---

## VI. Some strong points – business point of view (2/3)

- Multi-currency
- Management of the simultaneous sell of the the opening of a new credit and of the rise on the credit limit of another account
- Any conceivable credit product/offer may be created in a few clicks, by putting together pre-existent components
- Collateral management.
- Management of multiple “balance segments” with different “credit limits” inside one revolving credit account.
- Features making it possible for the operator to commit himself in the earlier phases of the selling process, without incurring the risk of a later rejection (“pre-grant”)
- Management of the pooling and consistency of granting rules between :
  - Selling an installment loan/selling a revolving credit/selling a package of many credit accounts
  - Selling a new credit/selling a limit rise on an existing account/selling a new funding on an existing account
  - Proposal and final grant
  - Selling process and personal guarantee process...

# LOAN OBJECTS

---

## VI. Some strong points – business point of view (3/3)

- Management of any kind of loan (credit cards, mortgage loans, customer accounts, purchased receivables, leasing ...)
- Base periodicity may be weekly, bi-weekly, monthly, bi-monthly ...
- Any payment schedule may be made up of any number of stages, each one with a different interest rate, amount, length, periodicity and progressiveness
- Management of more than one “direct debit” standard (local + SEPA...)
- Management of collection processes handling simultaneously all accounts of a customer.
- Design allowing the software to be used 7 days in a week and 24 h a day (the daily batch processing does not necessarily lock all accounts together)
- Basel 2-, IAS-, Sharia- and Obama’s 2009 credit card bill-compliant.
- The consistency between balances and entries is checked after **each** financial event.
- Extraction of end-of-month balances (for reporting needs) may be delayed till all late entries are made  
So “adjustment” accounting entries are no longer necessary.

# LOAN OBJECTS

---

## VII. Some other strong points – IT point of view

- *LOAN OBJECTS* deals only with the “business” level
  - The “technical” side of the full application may be organized freely
- The syntax used in *LOAN OBJECTS* (UML) is universally acknowledged to be the most suited to the building of “components” able to fit into any front-end software
- *LOAN OBJECTS*’ Concepts are always specialized “only when strictly necessary”, so that they might be re-used for other needs.

For instance :

80% of *LOAN OBJECTS* could be re-used to manage contracts of a nature other than Credit

100% of the “Account management” part could be re-used to manage Corporate Credit or even deposit accounts ...

→So any software designed on the basis of *LOAN OBJECTS* could easily be upgraded to a **full banking platform**