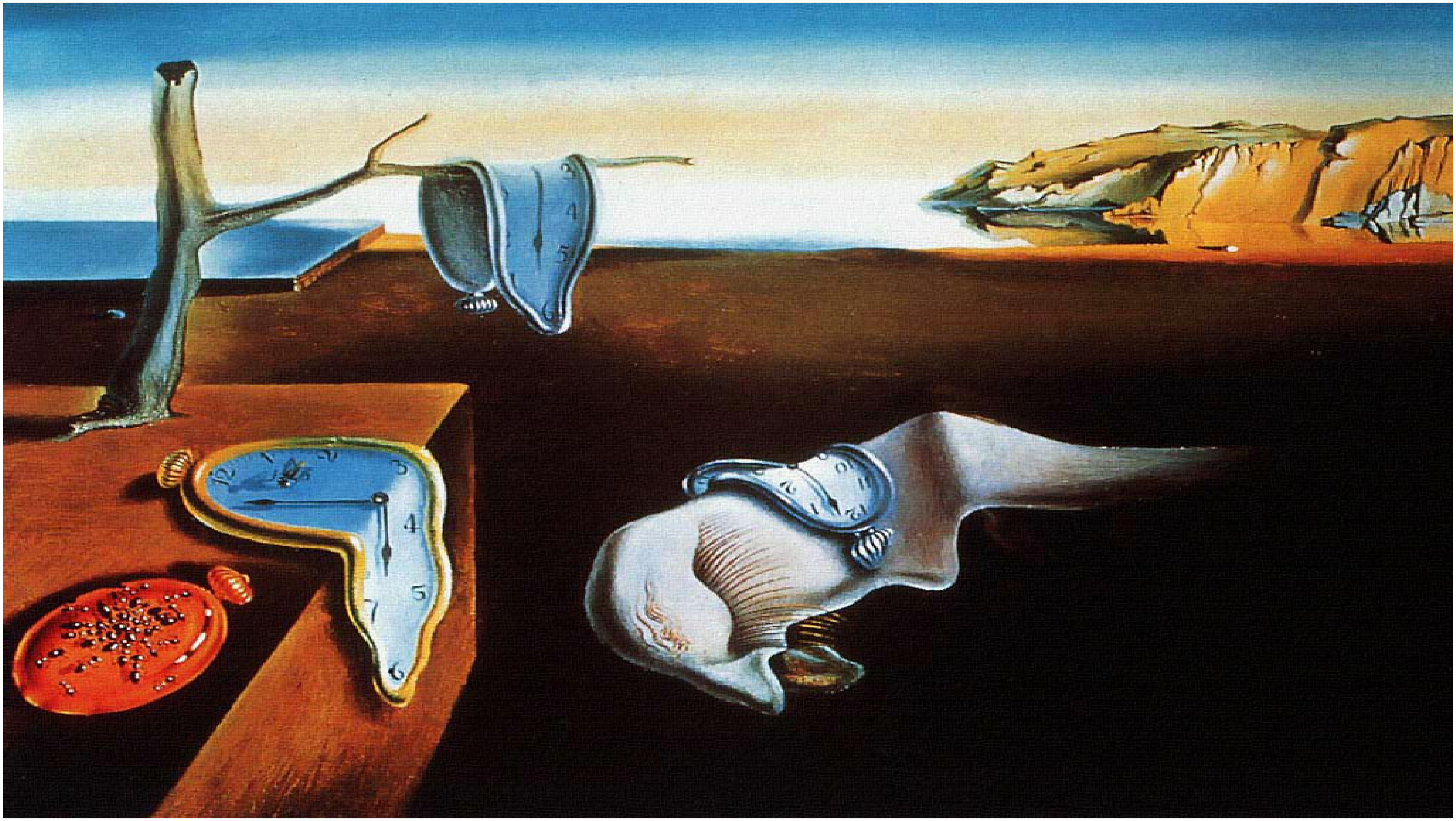


System Generation for Time and Activity Management Product Lines

presented by Jenya Levin

Ottawa-Carleton Institute for Computer Science

Overview

- Modeling and product lines
- Product line derivation
- Technologies involved
- Case studies
  1. Klok
  2. Leia
  3. Anuko Time Tracker
  4. TimeTrex
- Methodology analysis
- Future work
- Contributions
- References

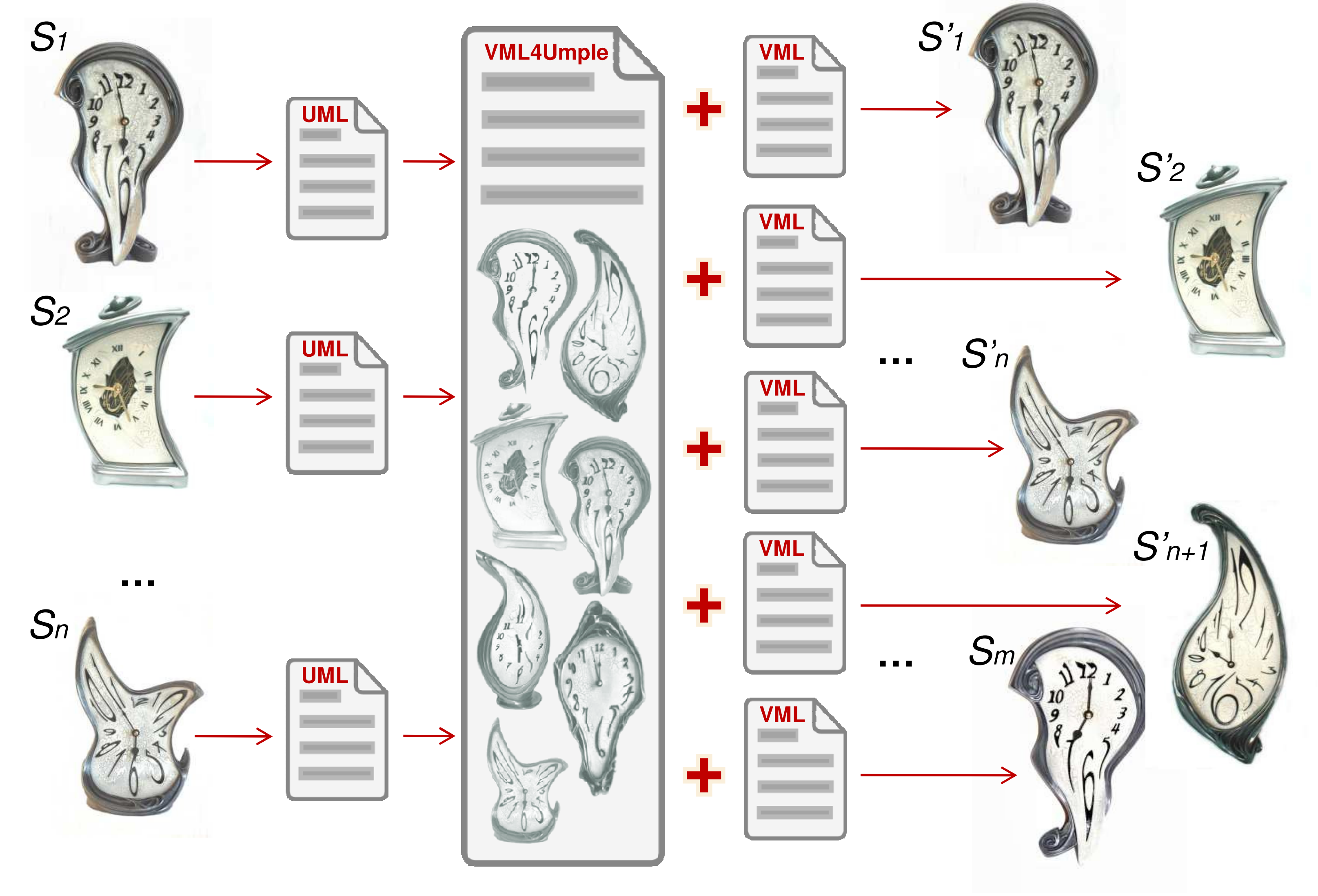
Modeling and Product Lines

- Modeling
  - Stakeholder communication and documentation
  - Modeling-driven development
  - Automated processing and code generation

- Product Lines
  - Commonalities and variabilities
  - Asset re-use and design for re-use
  - Quality control and regression testing
  - Documentation and traceability

Product Line Derivation

Original systems → *System models* → *Product line* + *Invocation files* → *Variety of systems in the domain*



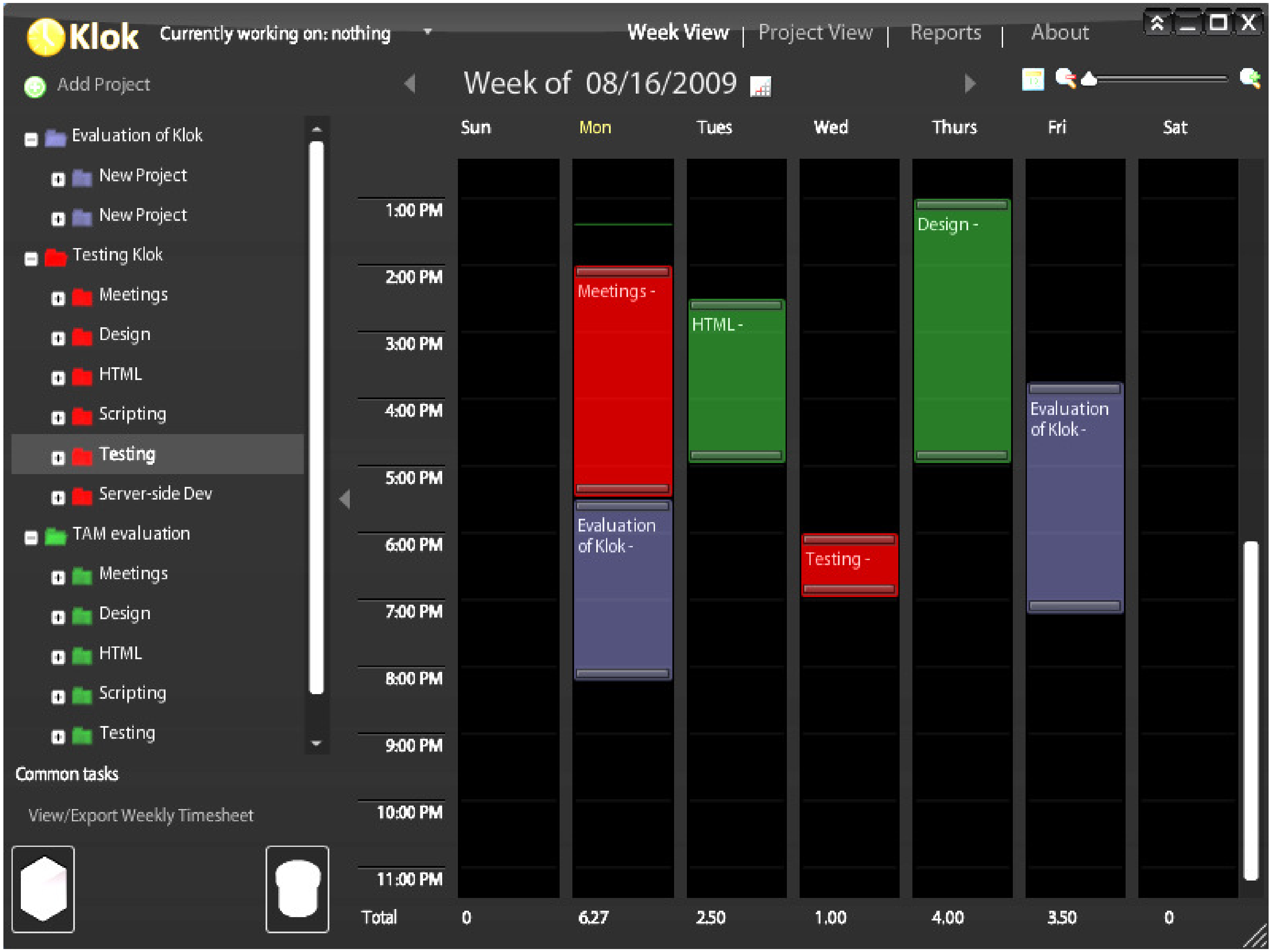
Technologies involved

- ERD – to extract application data structures
- UML – class and use case diagrams to model original systems
- UMLet – GUI tool for graphical UML modeling that uses XML-based file format
- Umple – textual language based on UML allowing for object-oriented code generation
- VML – language for modeling variabilities and invoking features to build individual systems
- VML4Umple – product line modeling language

Case Studies

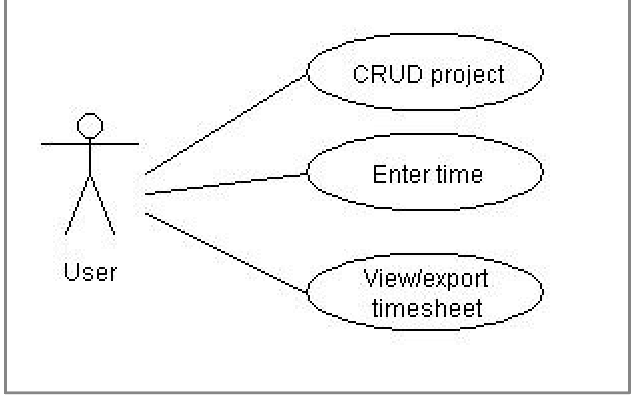
- Klok
  - Free, single-user
  - DB tables: N/A
  - Classes: 2
  - Clusters: 2
- Anuko Time Tracker
  - Open source, multi-user
  - DB tables: 16
  - Classes: 15
  - Clusters: 6
- Leia
  - Proprietary, multi-user
  - DB tables: 55
  - Classes: 54
  - Clusters: 7
- TimeTrex
  - Open source, multi-user
  - DB tables: 99
  - Classes: 122
  - Clusters: 17

Example – Klok - Screenshot

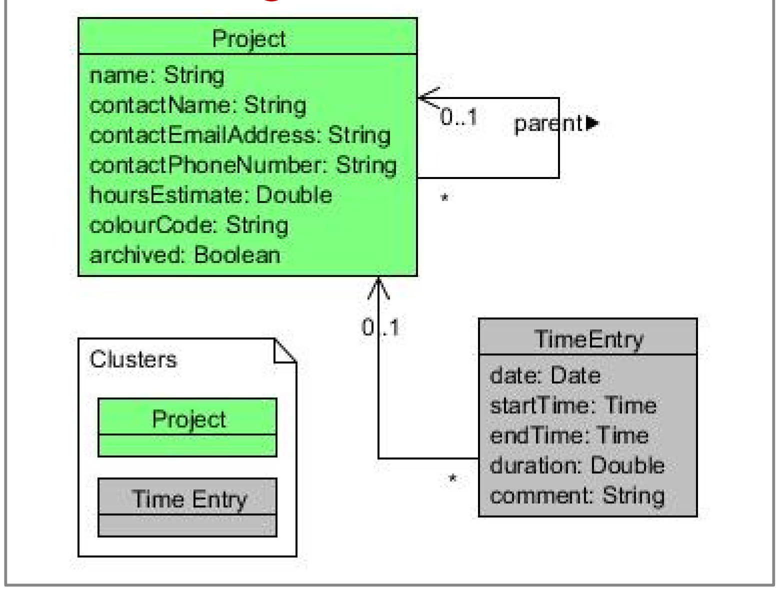


Example – Klok

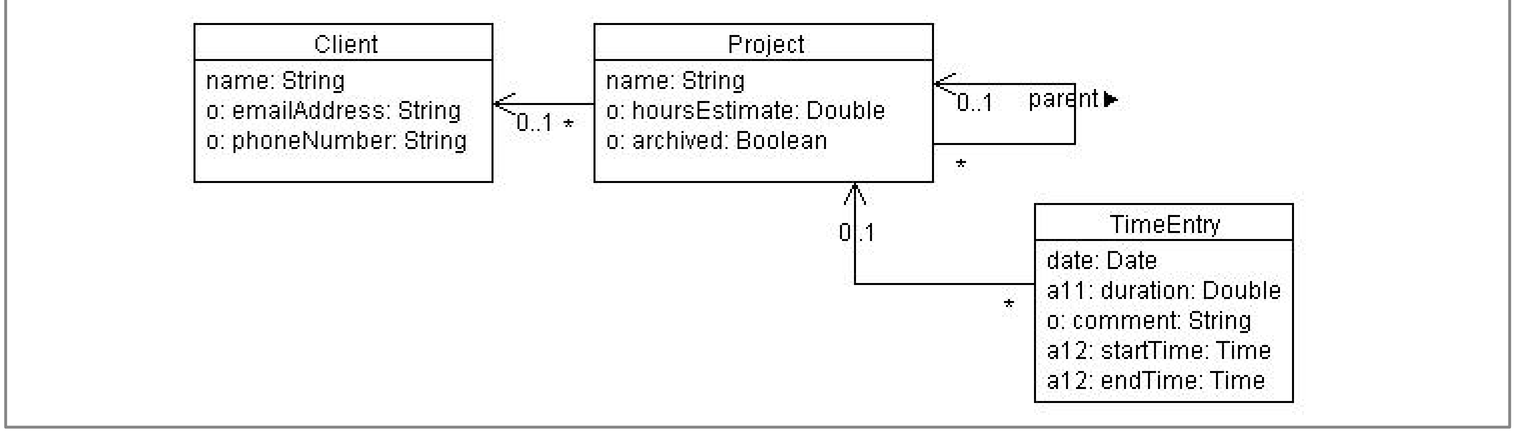
Use cases



Class diagram



Product line



Example – Klok – Umple code

```
class Project{
    String name;
    Double hoursEstimate;
    Boolean archived;
}
class TimeEntry{
    Date date;
    Double duration;
    String comment;
    Time startTime;
    Time endTime;
}
class Client{
    String name;
    String emailAddress;
    String phoneNumber;
}
association {
    0..1 Project parent <- * Project;
}
association {
    0..1 Project <- * TimeEntry;
}
association {
    0..1 Client <- * Project;
}
```

Example – VML4Umple

```
Concern CTimeEntry{  // time entry can have a rejection comment
    VariationPoint VPTimeEntryRejectedComment{
        Kind: Optional;
        class TimeEntry{
            String rejectedComment;
        }
    }
    // either duration or both start and end time are required
    VariationPoint VPEntryDuration{
        Kind: Alternative;
        Variant VDuration{
            class TimeEntry{
                 Double duration;
            }
        }
        Variant VStartEndTime{
            class TimeEntry{
                 Time startTime;
                 Time endTime;
            }
        }
    }
}
```

Example – Klok – Invocation file

```
// Invocation of a system similar to Klok

// Time Entry
// log time start and end times
invoke(CTimeEntry, VPEntryDuration, VStartEndTime);

// Project
// enter time against projects (client-related work items)
invoke(CTimeEntryAgainstProject, VPEntryAgainstProject, VProject);
// store optional comments for time entries
invoke(CTimeEntryAgainstProject, VPTimeEntryComment);
// allow projects to have parent projects
invoke(CTimeEntryAgainstProject, VPProjectParent);
// store project time estimates
invoke(CTimeEntryAgainstProject, VPProjectEstimate);
// allow archiving projects
invoke(CTimeEntryAgainstProject, VPProjectArchive);
// associate projects with clients
invoke(CTimeEntryAgainstProject, VPClient);
// store client email
invoke(CTimeEntryAgainstProject, VPClientEmail);
// store client phone number
invoke(CTimeEntryAgainstProject, VPClientPhoneNumber);
```

alternative variation points

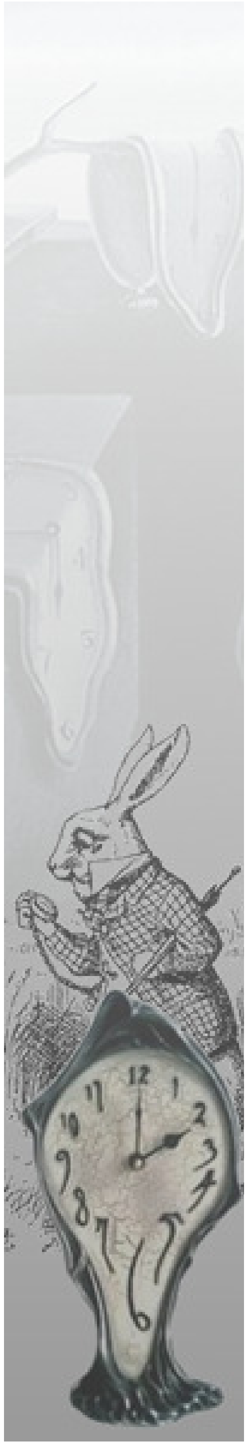
optional variation points

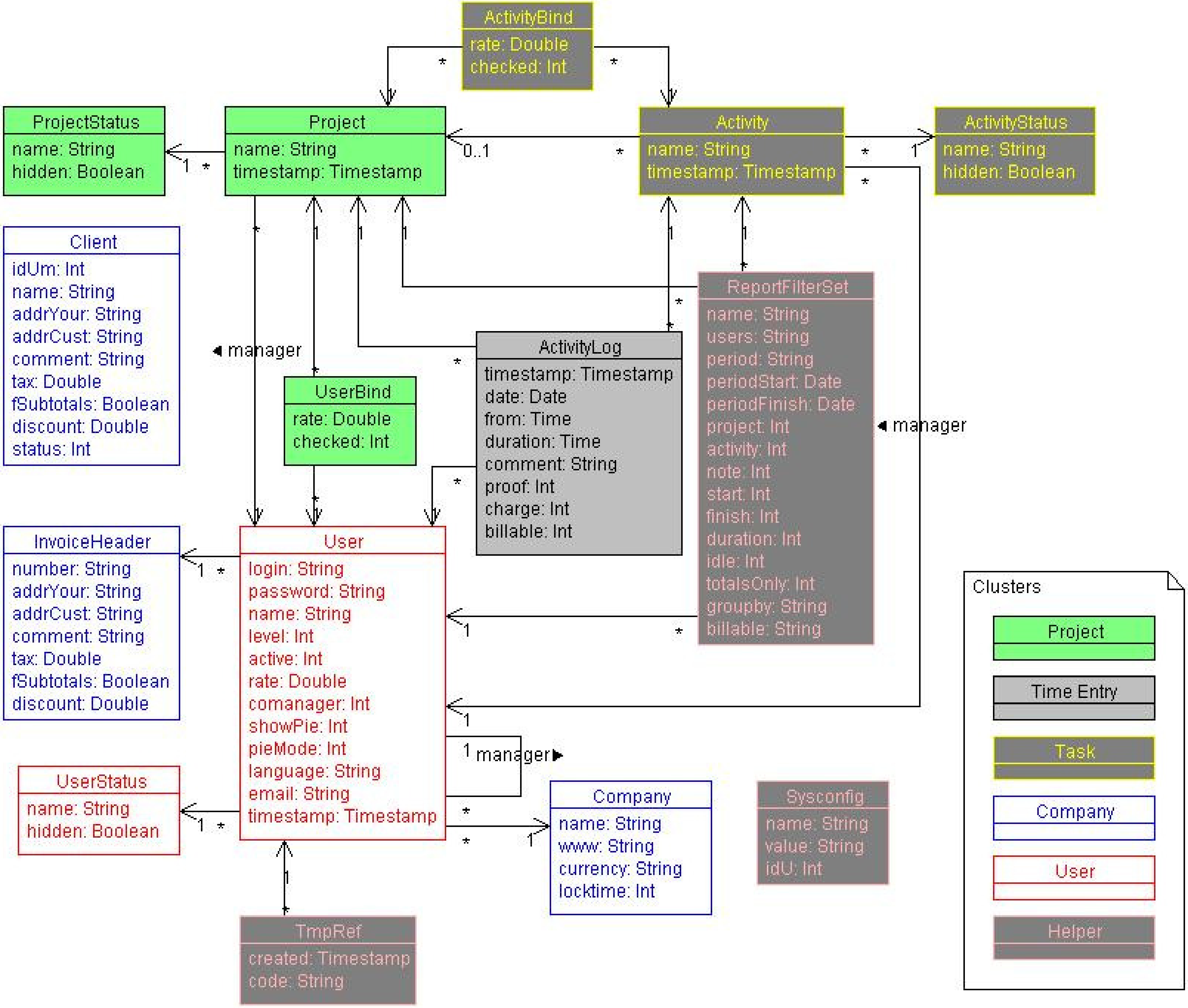
Example – Klok – Original vs invoked

```
// Umple code for original system
class Project{
    String name;
    Double hoursEstimate;
    Boolean archived;
}
class TimeEntry{
    Date date;
    Double duration;
    String comment;
    Time startTime;
    Time endTime;
}
class Client{
    String name;
    String emailAddress;
    String phoneNumber;
}
association {
    0..1 Project parent <- * Project;
}
association {
    0..1 Project <- * TimeEntry;
}
association {
    0..1 Client <- * Project;
}
```

```
// Umple code for the system based on Klok

class TimeEntry{ Date date; }
class TimeEntry{
    Time startTime;
    Time endTime;
}
class Project{ String name; }
association {
    0..1 Project <- * TimeEntry;
}
class TimeEntry{ String comment; }
association {
    0..1 Project parent <- * Project;
}
class Project{ Double timeEstimate; }
class Project{ Boolean archived; }
class Client{ String name; }
association {
    0..1 Client <- * Project;
}
class Client{ String emailAddress; }
class Client{ String phoneNumber; }
```





ActivityBind
rate: Double
checked: Int

ProjectStatus
name: String
hidden: Boolean

Project
name: String
timestamp: Timestamp

Activity
name: String
timestamp: Timestamp

ActivityStatus
name: String
hidden: Boolean

Client
idUm: Int
name: String
addrYour: String
addrCust: String
comment: String
tax: Double
fSubtotals: Boolean
discount: Double
status: Int

UserBind
rate: Double
checked: Int

ActivityLog
timestamp: Timestamp
date: Date
from: Time
duration: Time
comment: String
proof: Int
charge: Int
billable: Int

ReportFilterSet
name: String
users: String
period: String
periodStart: Date
periodFinish: Date
project: Int
activity: Int
note: Int
start: Int
finish: Int
duration: Int
idle: Int
totalsOnly: Int
groupby: String
billable: String

InvoiceHeader
number: String
addrYour: String
addrCust: String
comment: String
tax: Double
fSubtotals: Boolean
discount: Double

User
login: String
password: String
name: String
level: Int
active: Int
rate: Double
comanager: Int
showPie: Int
pieMode: Int
language: String
email: String
timestamp: Timestamp

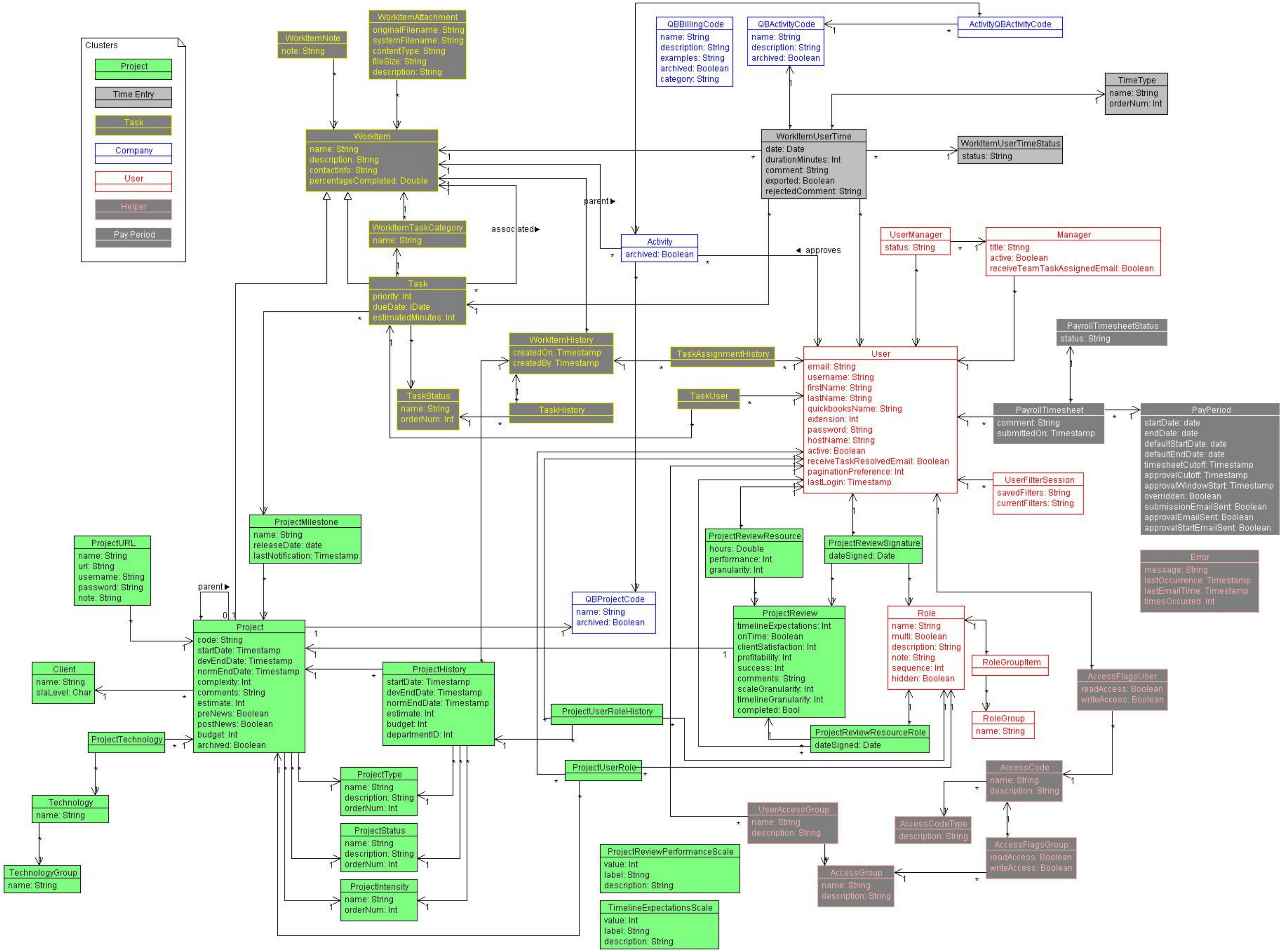
UserStatus
name: String
hidden: Boolean

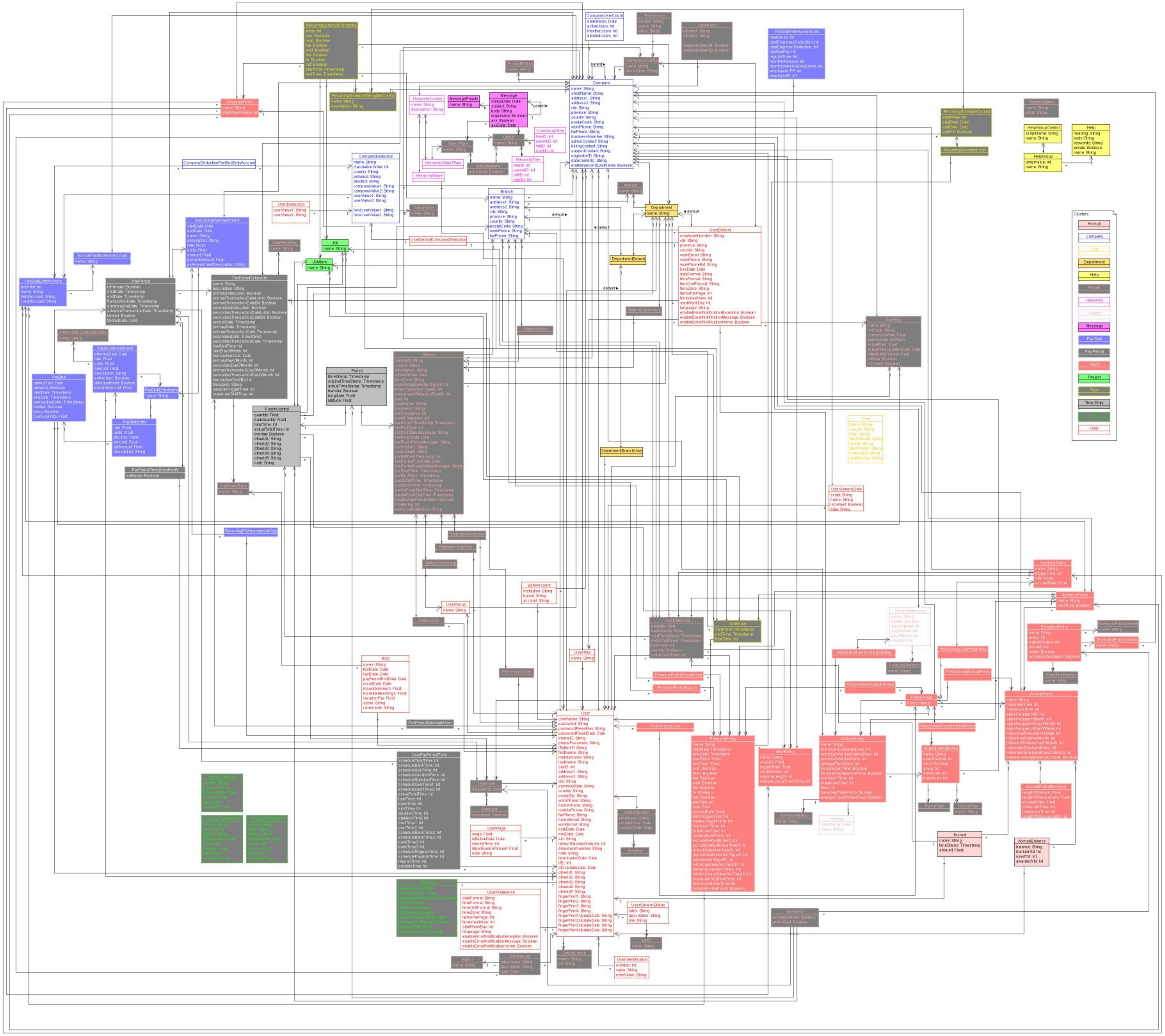
Company
name: String
www: String
currency: String
locktime: Int

Sysconfig
name: String
value: String
idU: Int

TmpRef
created: Timestamp
code: String

Clusters
Project
Time Entry
Task
Company
User
Helper


Clusters
Project
Time Entry
Task
Company
User
Helper
Pay Period
WorkItemNote
note: String
WorkItemAttachment
originalFilename: String
systemFilename: String
contentType: String
fileSize: String
description: String
WorkItem
name: String
description: String
contactInfo: String
percentageCompleted: Double
WorkItemTaskCategory
name: String
Task
priority: Int
dueDate: IDate
estimatedMinutes: Int
associated ▸
parent ▸
QBBillingCode
name: String
description: String
examples: String
archived: Boolean
category: String
QBActivityCode
name: String
description: String
archived: Boolean
ActivityQBActivityCode
TimeType
name: String
orderNum: Int
WorkItemUserTime
date: Date
durationMinutes: Int
comment: String
exported: Boolean
rejectedComment: String
WorkItemUserTimeStatus
status: String
Activity
archived: Boolean
◂ approves
UserManager
status: String
Manager
title: String
active: Boolean
receiveTeamTaskAssignedEmail: Boolean
PayrollTimesheetStatus
status: String
WorkItemHistory
createdOn: Timestamp
createdBy: Timestamp
TaskAssignmentHistory
TaskUser
TaskStatus
name: String
orderNum: Int
TaskHistory
User
email: String
username: String
firstName: String
lastName: String
quickbooksName: String
extension: Int
password: String
hostName: String
active: Boolean
receiveTaskResolvedEmail: Boolean
paginationPreference: Int
lastLogin: Timestamp
PayrollTimesheet
comment: String
submittedOn: Timestamp
PayPeriod
startDate: date
endDate: date
defaultStartDate: date
defaultEndDate: date
timesheetCutoff: Timestamp
approvalCutoff: Timestamp
approvalWindowStart: Timestamp
overridden: Boolean
submissionEmailSent: Boolean
approvalEmailSent: Boolean
approvalStartEmailSent: Boolean
UserFilterSession
savedFilters: String
currentFilters: String
Error
message: String
lastOccurrence: Timestamp
lastEmailTime: Timestamp
timesOccurred: Int
ProjectMilestone
name: String
releaseDate: date
lastNotification: Timestamp
ProjectURL
name: String
url: String
username: String
password: String
note: String
parent ▸
Project
code: String
startDate: Timestamp
devEndDate: Timestamp
normEndDate: Timestamp
complexity: Int
comments: String
estimate: Int
preNews: Boolean
postNews: Boolean
budget: Int
archived: Boolean
QBProjectCode
name: String
archived: Boolean
ProjectReviewResource
hours: Double
performance: Int
granularity: Int
ProjectReviewSignature
dateSigned: Date
ProjectReview
timelineExpectations: Int
onTime: Boolean
clientSatisfaction: Int
profitability: Int
success: Int
comments: String
scaleGranularity: Int
timelineGranularity: Int
completed: Bool
Role
name: String
multi: Boolean
description: String
note: String
sequence: Int
hidden: Boolean
RoleGroupItem
RoleGroup
name: String
AccessFlagsUser
readAccess: Boolean
writeAccess: Boolean
Client
name: String
slaLevel: Char
ProjectHistory
startDate: Timestamp
devEndDate: Timestamp
normEndDate: Timestamp
estimate: Int
budget: Int
departmentID: Int
ProjectUserRoleHistory
ProjectReviewResourceRole
dateSigned: Date
ProjectTechnology
ProjectUserRole
Technology
name: String
TechnologyGroup
name: String
ProjectType
name: String
description: String
orderNum: Int
ProjectStatus
name: String
description: String
orderNum: Int
ProjectIntensity
name: String
orderNum: Int
UserAccessGroup
name: String
description: String
AccessCode
name: String
description: String
AccessCodeType
description: String
AccessFlagsGroup
readAccess: Boolean
writeAccess: Boolean
AccessGroup
name: String
description: String
ProjectReviewPerformanceScale
value: Int
label: String
description: String
TimelineExpectationsScale
value: Int
label: String
description: String
0..1
1
*


RecurringScheduleTemplate
week: Int
sun: Boolean
mon: Boolean
tue: Boolean
wed: Boolean
thu: Boolean
fri: Boolean
sat: Boolean
startTime: Timestamp
endTime: Timestamp
SchedulePolicy
name: String
startStopWindow: Int
RecurringScheduleTemplateControl
name: String
description: String
CompanyDeductionPayStubEntryAccount
CompanyDeduction
name: String
claculationOrder: Int
country: String
province: String
disctrict: String
companyValue1: String
companyValue2: String
userValue1 : String
userValue2: String
...
lockUserValue1 : String
lockUserValue2: String
UserDeduction
userValue1 : String
userValue2: String
Calculation
name: String
UserDefaultCompanyDeduction
RecurringPSAmendment
startDate: Date
endDate: Date
name: String
description: String
rate: Float
units: Float
amount: Float
percentAmount: Float
psAmendmentDescription: String
StartWeekDay
name: String
Job
name: String
JobItem
name: String
AccrualPayStubEntryAccount
name: String
PayStubEntryAccount
psOrder: Int
name: String
debitAccount: String
creditAccount: String
PayPeriod
isPrimary: Boolean
startDate: Timestamp
endDate: Timestamp
transactionDate: Timestamp
advanceEndDate: Timestamp
advanceTransactionDate: Timestamp
tainted: Boolean
taintedDate: Date
PayPeriodSchedule
name: String
description: String
primaryDateLdom: Boolean
primaryTransactionDateLdom: Boolean
primaryTransactionDateBd: Boolean
secondaryDateLdom: Boolean
secondaryTransactionDateLdom: Boolean
secondaryTransactionDateBd: Boolean
anchorDate: Timestamp
primaryDate: Timestamp
primaryTransactionDate: Timestamp
secondaryDate: Timestamp
secondaryTransactionDate: Timestamp
dayStartTime: Int
startDayOfWeek: Int
transactionDate: Date
primaryDayOfMonth: Int
secondaryDayOfMonth: Int
primaryTransactionDayOfMonth: Int
secondaryTransactionDayOfMonth: Int
transactionDateBd: Int
timeZone: String
newDayTriggerTime: Int
maximumShiftTime: Int
PercentAmountEntryName
name: String
PayStubAmendment
effectiveDate: Date
rate: Float
units: Float
amount: Float
description: String
authorized: Boolean
ytdAdjustment: Boolean
percentAmount: Float
PayStubEntryName
name: String
PayStub
statusDate: Date
advance: Boolean
startDate: Timestamp
endDate: Timestamp
transactionDate: Timestamp
tainted: Boolean
temp: Boolean
currencyRate: Float
PayStubEntry
rate: Float
units: Float
ytdUnits: Float
amount: Float
ytdAmount: Float
description: String
PayPeriodTimeSheetVerify
authorize: Boolean
Punch
timeStamp: Timestamp
originalTimeStamp: Timestamp
actualTimeStamp: Timestamp
transfer: Boolean
longitude: Float
latitude: Float
PunchControl
quantity: Float
badQuantity: Float
totalTime: Int
actualTotalTime: Int
overlap: Boolean
otherId1: String
otherId2: String
otherId3: String
otherId4: String
otherId5: String
note: String
PayPeriodType
name: String
RecurringPSAmendmentUser
CompanyUserCount
dateStamp: Date
activeUsers: Int
inactiveUsers: Int
deletedUsers: Int
Permission
section: String
name: String
value: String
OtherField
otherId1 : String
otherId2: String
...
requiredOtherId1 : Boolean
requiredOtherId2: Boolean
...
PermissionControl
name: String
description: String
ProductEdition
name: String
parent ▸
Company
name: String
shortName: String
address1 : String
address2: String
city: String
province: String
country: String
postalCode: String
workPhone: String
faxPhone: String
businessNumber: String
adminContact: String
billingContact: String
supportContact: String
originatorID: String
dataCenterID: String
enableSecondLastName: Boolean
HierarchyControl
name: String
description: String
MessagePriority
name: String
Message
statusDate: Date
subject: String
body: String
requireAck: Boolean
ack: Boolean
ackDate: Date
*parent ▸
UserGroupTree
treeID: Int
parentID: Int
leftID: Int
rightID: Int
Object
name: String
ObjectType
name: String
HierarchyObjectType
HierarchyShare
Authorizations
authorized: Boolean
HierarchyTree
treeID: Int
parentID: Int
leftID: Int
rightID: Int
Branch
name: String
address1 : String
address2: String
city: String
province: String
country: String
postalCode: String
workPhone: String
faxPhone: String
default ▸
◂ default
Manual
name: String
Department
name: String
◂ default
UserDefault
employeeNumber: String
city: String
province: String
country: String
workEmail: String
workPhone: String
workPhoneExt: String
hireDate: Date
dateFormat: String
timeFormat: String
timeUnitFormat: String
timeZone: String
itemsPerPage: Int
timesheetView: Int
startWeekDay: Int
language: String
enableEmailNotificationException: Boolean
enableEmailNotificationMessage: Boolean
enableEmailNotificationHome: Boolean
DepartmentBranch
default ▸
StationDepartment
StationBranch
Station
stationID: String
source: String
description: String
allowedDate: Date
timeZone: String
userGroupSelectionTypeID: Int
branchSelectionTypeID: Int
departmentSelectionTypeID: Int
port: Int
userName: String
password: String
pollFrequency: Int
pushFrequency: Int
lastPunchTimeStamp: Timestamp
lastPollDate: Int
lastPollStatusMessage: String
lastPushDate: Date
lastPushStatusMessage: String
userValue1 : String
userValue5: String
partialPushFrequency: Int
lastPartialPushDate: Date
lastPartialPushStatusMessage: String
pullStartTime: Timestamp
pullEndTime: Timestamp
pushStartTime: Timestamp
pushEndTime: Timestamp
partialPushStartTime: Timestamp
partialPushEndTime: Timestamp
enableAutoPunchStatus: Boolean
modeFlag: Int
workCodeDefinition: String
DepartmentBranchUser
StationExcludeUser
StationIncludeUser
StationUserGroup
BankAccount
institution: String
transit: String
account: String
UserGroup
name: String
StationUser
PayStubEntryAccountLink
totalGross: Int
totalEmployeeDeduction: Int
totalEmployerDeduction: Int
totalNetPay: Int
regularTime: Int
monthlyAdvance: Int
monthlyAdvanceDeduction: Int
employeeCPP: Int
employeeEI: Int
RecurringScheduleControl
startWeek: Int
startDate: Date
endDate: Date
autoFill: Boolean
RecurringScheduleUser
SystemSetting
name: String
value: String
HelpGroupControl
scriptName: String
name: String
Help
heading: String
body: String
keywords: String
private: Boolean
name: String
HelpGroup
orderValue: Int
name: String
Clusters
Accrual
Company
Cron
Department
Help
Helper
Hierarchy
Holiday
Message
Pay Stub
Pay Period
Policy
Project
Task
Time Entry
Tax
User
Currency
name: String
isoCode: String
conversionRate: Float
autoUpdate: Boolean
actualRate: Float
actualRateUpdatedDate: Date
rateModifyPercent: Float
isBase: Boolean
isDefault: Boolean
Cron
name: String
minute: String
hour: String
dayOfMonth: String
month: String
dayOfWeek: String
command: String
lastRunDay: String
UserGenericData
script: String
name: String
isDefault: Boolean
data: String
OvertimePolicy
name: String
triggerTime: Int
rate: Float
accrualRate: Float
AbsencePolicy
name: String
overTime: Boolean
ExceptionPolicySeverity
name: String
ExceptionPolicyControl
name: String
ExceptionPolicy
name: String
grace: Int
watchWindow: Int
demerit: Int
active: Boolean
enableAuthorization: Boolean
EmailNotification
name: String
RecurringHoliday
name: String
easter: Boolean
weekInterval: Int
dayOfWeek: Int
dayOfMonth: Int
monthInt: Int
HolidayPolicyRecurringHoliday
PivotDayDirection
name: String
PolicyGroupOvertimePolicy
PolicyGroupAccrualPolicy
PolicyGroupPremiumPolicy
PolicyGroup
name: String
PolicyGroupRoundIntervalPolicy
AccrualPolicy
name: String
minimumTime: Int
maximumTime: Int
applyFrequencyID: Int
applyFrequencyMonth: Int
applyFrequencyDayOfMonth: Int
applyFrequencyDayOfWeek: Int
milestoneRolloverHireDate: Int
milestoneRolloverMonth: Int
milestonRolloverDayOfMonth: Int
minimumEmployedDays: Int
minimumEmployedDaysCatchup: Int
enablePayStubBalanceDisplay: Boolean
HolidayPolicy
name: String
minimumEmployedDays: Int
minimumWorkedPeriodDays: Int
minimumWorkedDays: Int
averageTimeDays: Int
includeOverTime: Boolean
includePaidAbsenceTime: Boolean
minimumTime: Int
maximumTime: Int
time: Int
forceOverTimePolicy: Boolean
averageTimeWorkedDays: Boolean
RoundIntervalPolicy
name: String
roundInterval: Int
strict: Boolean
grace: Int
minimum: Int
maximum: Int
PunchType
RoundType
name: String
AccrualPolicyMilestone
lengthOfService: Float
lengthOfServiceDays: Float
accrualRate: Float
minimumTime: Int
maximumTime: Int
lengthOfServiceUnitID: Int
Holiday
dateStamp: Date
name: String
ScheduleStatus
name: String
Accrual
name: String
timeStamp: Timestamp
amount: Float
AccrualBalance
balance: String
bankedYtd: Int
usedYtd: Int
awardedYtd: Int
UserDateTotal
quantity: Float
badQuantity: Float
startTimeStamp: Timestamp
endTimeStamp: Timestamp
totalTime: Int
override: Boolean
actualTotalTime: Int
Schedule
startTime: Timestamp
endTime: Timestamp
totalTime: Int
UserTitle
name: String
PremiumPolicyDepartment
PremiumPolicyBranch
PermissionUser
ROE
name: String
firstDate: Date
lastDate: Date
payPeriodEndDate: Date
recallDate: Date
insurableHours: Float
insurableEarnings: Float
vacationPay: Float
serial: String
comments: String
PayPeriodScheduleUser
UserPayPeriodTotal
scheduleTotalTime: Int
scheduleBankTime: Int
scheduleSickTime: Int
scheduleVacationTime: Int
scheduleStatutoryTime: Int
scheduleOverTime1 : Int
scheduleOverTime2: Int
actualTotalTime: Int
totalTime: Int
bankTime: Int
sickTime: Int
vacationTime: Int
statutoryTime: Int
overTime1: Int
overTime2: Int
scheduleBankTime2: Int
scheduleBankTime3: Int
bankTime2: Int
bankTime3: Int
scheduleRegularTime: Int
schedulePayableTime: Int
regularTime: Int
payableTime: Int
IncomeTaxRate
country: String
province: String
effectiveDate: Date
income: Float
rate: Float
constant: Float
IncomeTaxRateCR
country: String
state: String
district: String
effectiveDate: Date
status: Int
income: Float
rate: Float
constant: Float
IncomeTaxRateUS
country: String
state: String
district: String
effectiveDate: Date
status: Int
income: Float
rate: Float
constant: Float
UserTax
federalClaim: Float
provincialClaim: Float
federalAdditionalDeduction: Float
wcbRate: Float
eiExempt: Boolean
cppExempt: Boolean
federalTaxExempt: Boolean
provincialTaxExempt: Boolean
vacationRate: Float
releaseVacation: Boolean
UserDate
dateStamp: Date
Request
authorized: Boolean
UserWage
wage: Float
effectiveDate: Date
weeklyTime: Int
laborBurdenPercent: Float
note: String
UserPreference
dateFormat: String
timeFormat: String
timeUnitFormat: String
timeZone: String
itemsPerPage: Int
timesheetView: Int
startWeekDay: Int
language: String
enableEmailNotificationException: Boolean
enableEmailNotificationMessage: Boolean
enableEmailNotificationHome: Boolean
Action
name: String
SystemLog
tableName: String
description: String
date: Date
User
userName: String
password: String
passwordResetKey: String
passwordResetDate: Date
phoneID: String
phonePassword: String
iButtonID: String
firstName: String
middleName: String
lastName: String
sexID: Int
address1 : String
address2: String
city: String
provinceState: String
country: String
postalZip: String
workPhone: String
homePhone: String
mobilePhone: String
faxPhone: String
homeEmail: String
workEmail: String
birthDate: Date
hireDate: Date
sin: String
labourStandardIndustry: Int
employeeNumber: String
note: String
terminationDate: Date
rfID: Int
rfIDUpdateDate: Date
otherId1 : String
otherId2: String
otherId3: String
otherId4: String
otherId5: String
fingerPrint1 : String
fingerPrint2: String
fingerPrint3: String
fingerPrint4: String
fingerPrint1UpdateDate: String
fingerPrint2UpdateDate: String
fingerPrint3UpdateDate: String
fingerPrint4UpdateDate: String
PolicyGroupUser
Authentication
ipAddress: String
createdDate: Date
updatedDate: Date
Session
UserGenericStatus
label: String
description: String
link: String
Batch
name: String
BreadCrumb
name: String
url: String
UserIdentification
number: Int
value: String
extraValue: String
PremiumPolicy
name: String
startDate: Timestamp
endDate: Timestamp
startTime: Time
endTime: Time
sun: Boolean
mon: Boolean
tue: Boolean
wed: Boolean
thu: Boolean
fri: Boolean
sat: Boolean
payType: Int
rate: Float
accrualRate: Float
dailyTriggerTime: Int
weeklyTriggerTime: Int
minimumTime: Int
maximumTime: Int
includeMealPolicy: Int
excludeDefaultBranch: Int
excludeDefaultDepartment: Int
branchSelectionTypeID: Int
departmentSelectionTypeID: Int
jobSelectionTypeID: Int
jobGroupSelectionTypeID: Int
jobItemSelectionTypeID: Int
jobItemGroupSelectionTypeID: Int
maximumNoBreakTime: Int
minimumBreakTime: Int
includePartialPunch: Boolean
MealPolicy
name: String
amount: Float
triggerTime: Time
startWindow: Int
windowLength: Int
includeLunchPunchTime: Int
Exception
enableDemerit: Boolean
authorized: Boolean

Methodology Analysis

- Domain-specific product line derivation:
  1. Analyze and model several existing applications
  2. Iteratively bring the systems to a common base
  3. Build product line from base case up

- Process automation
  - First two steps require human involvement
  - Mapping similar functionality elements
  - Annotations for original systems

Future Work

- Generate original systems through annotations
- Feature selection via dependency tree
- GUI-driven invocation file adjustments
- GUI-driven system addition to product line
- Umple-based UI generation for CRUD functions
- Product lines for other domains
  - Registration systems
  - Shopping carts and point-of-sale systems
  - Blogs and forums
  - Task management and scheduling
  - Calendars
  - Budget applications

Contributions

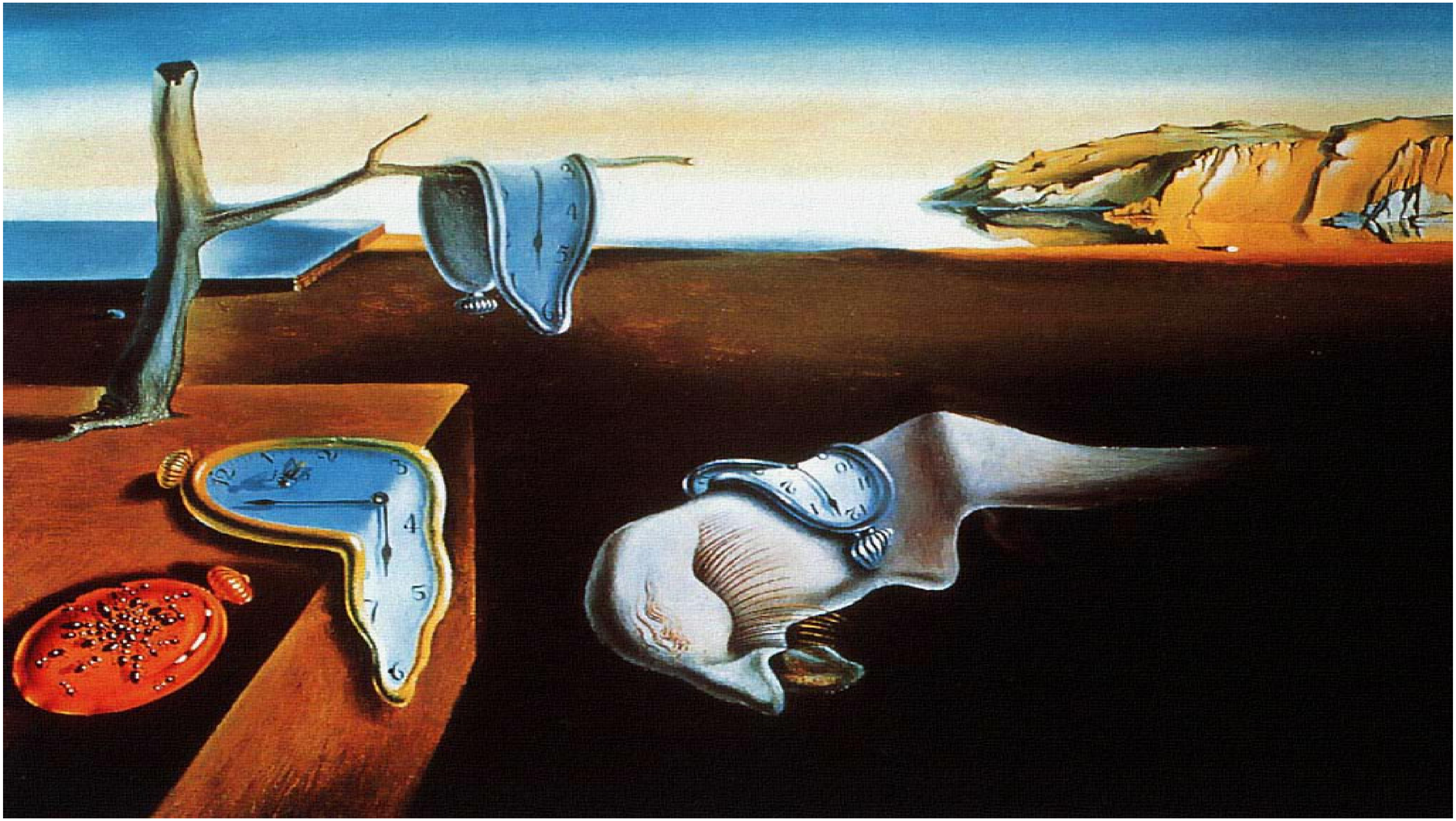
- General methodology for product line derivation
  - Thoroughly documented derivation process
  - Case study in TAM domain
  - VML4Umple language
- Time and activity product line
  - Models and generated code
  - Suggestions on automation
  - Possible future extensions

References

- Auer, M., Tschurtschenthaler, T. and Biffl, S. (2003, *A flyweight UML modelling tool for software development in heterogeneous environments. Euromicro Conference, 2003. Proceedings. 29th*, pp. 267-272, 2003.
- Forward, A., Lethbridge, T. C. and Brestovansky, D. *Improving program comprehension by enhancing program constructs: An analysis of the Umple language*, in 2009, pp. 311-312.
- Loughran, N., Sánchez, P., Garcia, A. and Fuentes, L. (2008, *Software Composition*, pp. 36-51, 2008.
- *System Generation for Time and Activity Management Product Lines - Support materials*, accessed 2009, http://www.site.uottawa.ca/~tcl/gradtheses/jlevin/
- *Software Architecture: Foundations, Theory, and Practice*. Wiley Publishing, 2009, pp. 750.
- *VML Online*, accessed 2009, http://cruise.site.uottawa.ca/umpleonline/vml.html

References

- Sánchez, P., Loughran, N., Fuentes, L. and Garcia, A., "Engineering languages for specifying product-derivation processes in software product lines," in *Software Language Engineering: First International Conference, SLE 2008, Toulouse, France, September 29-30, 2008. Revised Selected Papers* Anonymous Springer-Verlag, 2009, pp. 188-207.
- *Klok*, accessed 2009, http://klok.mcgraphix.com/
- *Lixar I.T. Inc.*, accessed 2009, http://www.lixar.com/
- *Anuko TimeTracker*, accessed 2009, http://www.anuko.com/content/time_tracker/
- *TimeTrex*, accessed 2009, http://www.timetrex.com/
- Illustration References
  1. Salvador Dalí, *La persistencia de la memoria*, 1931
  2. John Tenniel , *White Rabbit*, 1866, illustration to *Alice's Adventures in Wonderland* by Lewis Carroll, 1865
  3. Arabesque gifts, Dalí-style melting clocks, http://www.arabesque-gifts.co.uk/, 2009



Thank you

Questions?