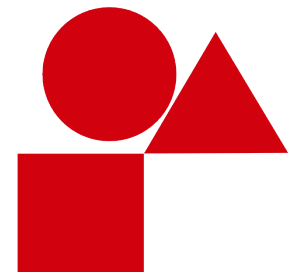


KinOath Kinship Archiver Version 1.4

The kinship archiving software under development by Peter Withers at the Language Archive, MPI, Nijmegen



Introduction

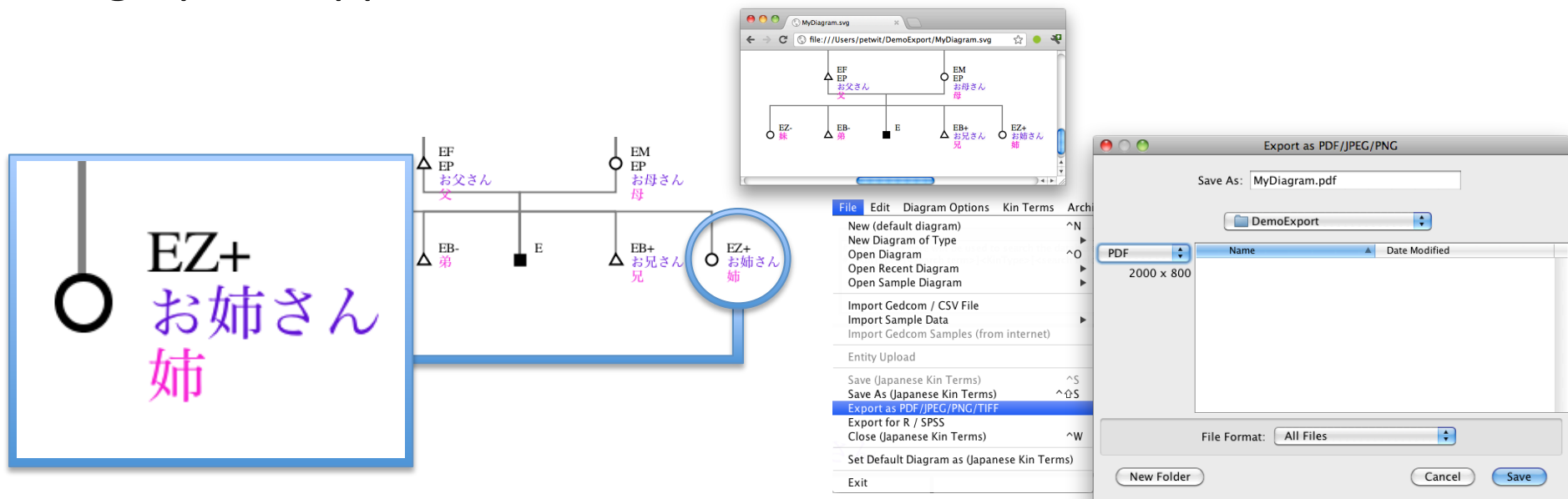
- KinOath Kinship Archiver is a kinship application under development by Peter Withers at the TLA language archive.
- Its primary goal is to connect kinship data with archived data, such as audio, video or written resources while also being closely integrated with the archive software such as Arbil.
- Beyond this goal it is designed to be flexible and culturally nonspecific, such that culturally different social structures can equally be represented.
- This talk will show examples of how the current version of the desktop application can be used and also discuss features that are in development.

Current Release

- The fifth stable version 1.4 is now available.
- The future release 1.5 has been translated into a few languages and this will be available in the near future.
- We are considering the use of LaunchPad to allow community driven translations in the future.
- An updated manual is available in the application and on the website.
- The testing versions usually available, if you want to experiment with the untested features.
- The changes list is available on the download page.
- The todo list is also available on the download page.
- <http://tla.mpi.nl/tools/tla-tools/kinoath>

Publishable Diagrams

- All the diagrams produced are in a scalable vector graphics (SVG) format of publishing quality
- All diagrams can be exported into PDF format
- The working files are in SVG format that can be opened in graphics applications or viewed in a web browser



Kin Type Strings

- Kin Type strings are a shorthand way to describe kinship relations
- For example:
 - M indicates a mother relation
 - F indicates a father relation
- There are a number of variants of this shorthand
- KinOath offers a set of commonly used defaults
- But the user can customise these to their preferred shorthand

Kin Type Strings in KinOath

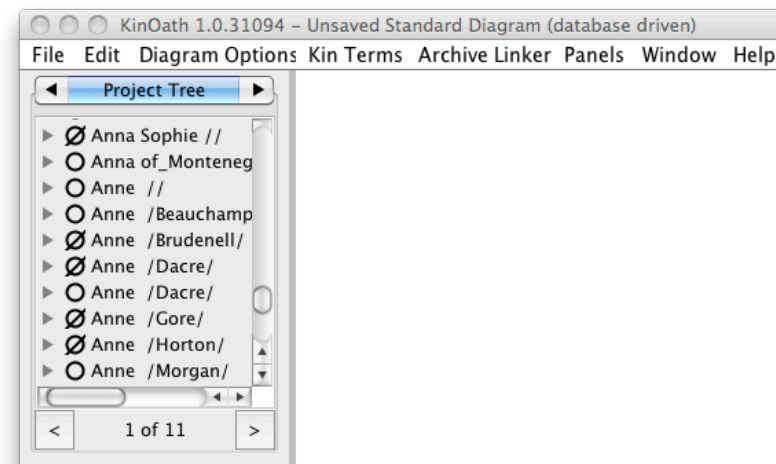
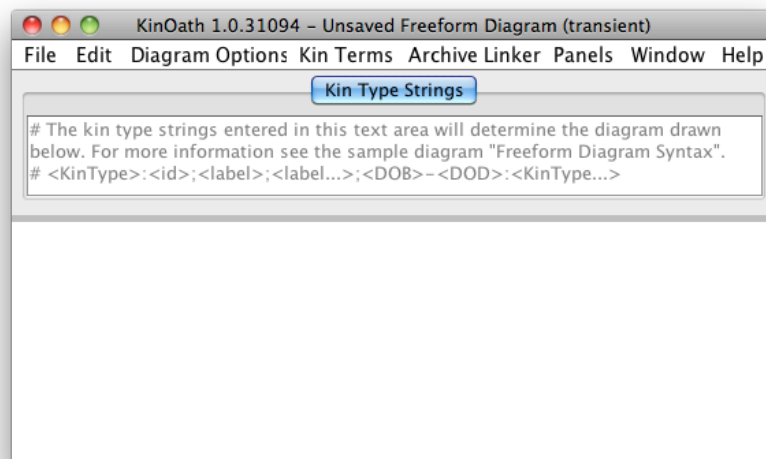
- Kin type strings are used throughout the application
- They are used to search kinship data and to generate diagrams
- These kin type strings can be customized if required
- Each kin type can use any string, any relation type and any symbol
- These kin types are defined and stored in each diagram file
- They can also be stored in the default diagram of a project

Kin Type String	Relation Type	Symbol Type	Display Name	
Ef	none	circle	Ego Female	<input type="checkbox"/>
Em	none	triangle	Ego Male	<input type="checkbox"/>
Fa	ancestor	triangle	Father	<input type="checkbox"/>
Mo	ancestor	circle	Mother	<input type="checkbox"/>
Br	sibling	triangle	Brother	<input type="checkbox"/>
Si	sibling	circle	Sister	<input type="checkbox"/>
So	descendant	triangle	Son	<input type="checkbox"/>
Da	descendant	circle	Daughter	<input type="checkbox"/>
Hu	union	triangle	Husband	<input type="checkbox"/>
Wi	union	circle	Wife	<input type="checkbox"/>
Pa	ancestor	square	Parent	<input type="checkbox"/>
Sb	sibling	square	Sibling	<input type="checkbox"/>
Sp	union	square	Spouse	<input type="checkbox"/>
Ch	descendant	square	Child	<input type="checkbox"/>
F	ancestor	triangle	Father	<input type="checkbox"/>
M	ancestor	circle	Mother	<input type="checkbox"/>
B	sibling	triangle	Brother	<input type="checkbox"/>
Z	sibling	circle	Sister	<input type="checkbox"/>
S	descendant	triangle	Son	<input type="checkbox"/>
D	descendant	circle	Daughter	<input type="checkbox"/>
H	union	triangle	Husband	<input type="checkbox"/>
W	union	circle	Wife	<input type="checkbox"/>
P	ancestor	square	Parent	<input type="checkbox"/>
G	sibling	square	Sibling	<input type="checkbox"/>
E	none	square	Ego	<input type="checkbox"/>
C	descendant	square	Child	<input type="checkbox"/>
m	none	triangle	Male	<input type="checkbox"/>
f	none	circle	Female	<input type="checkbox"/>
x	none	square	Undefined	<input type="checkbox"/>
*			Any Relation	<input type="checkbox"/>

F	ancestor	triangle	Father
M	ancestor	circle	Mother
B	sibling	triangle	Brother
Z	sibling	circle	Sister
S	descendant	triangle	Son
D	descendant	circle	Daughter
H	union	triangle	Husband
W	union	circle	Wife
P	ancestor	square	Parent
G	sibling	square	Sibling
E	none	square	Ego
C	descendant	square	Child

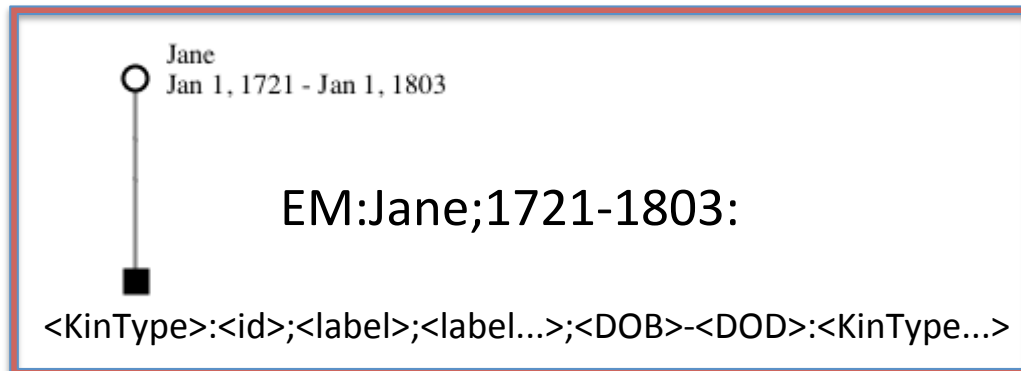
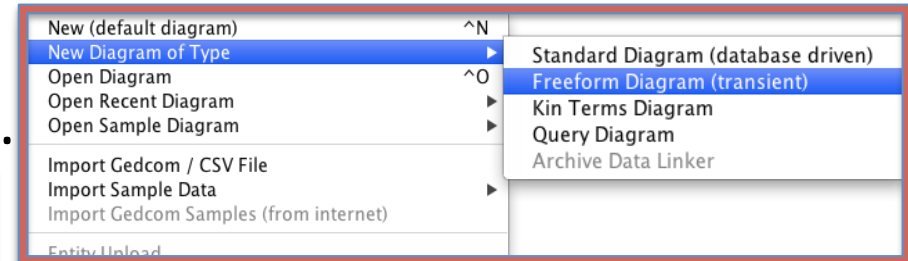
Freeform vs Project

- Freeform diagrams have no kinship data records.
- The data only exists on the diagram.
- The kin type strings or kin term definitions are the only source data for the diagram.
- Project diagrams display and query kinship records stored in a project.
- There are kin data files for each entity.
- Changes to kin data is reflected on all diagrams using that project.
- Data can be imported from GEDCOM or CSV.

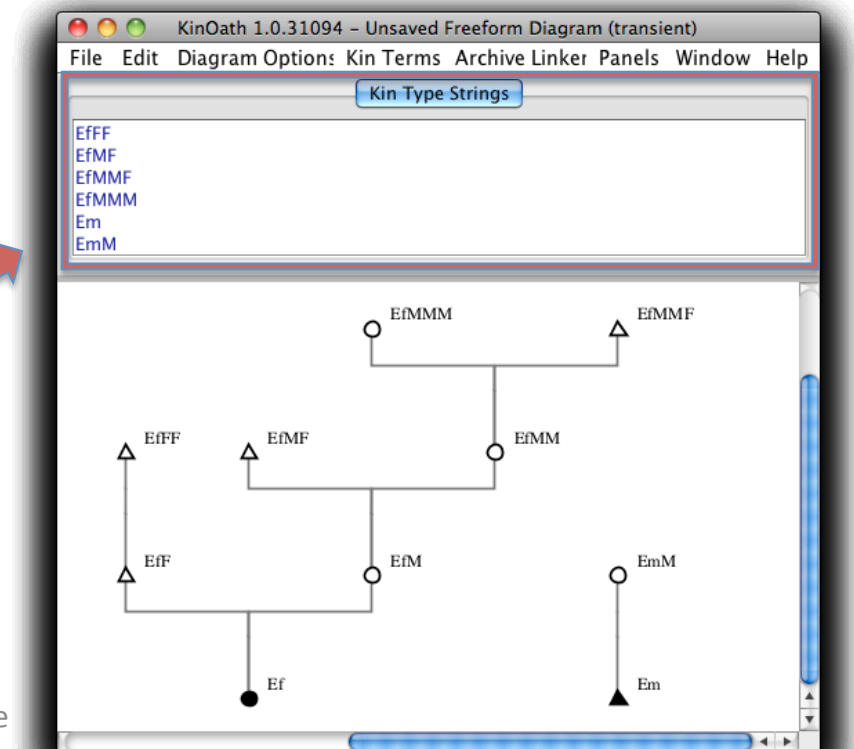


Freeform Diagram

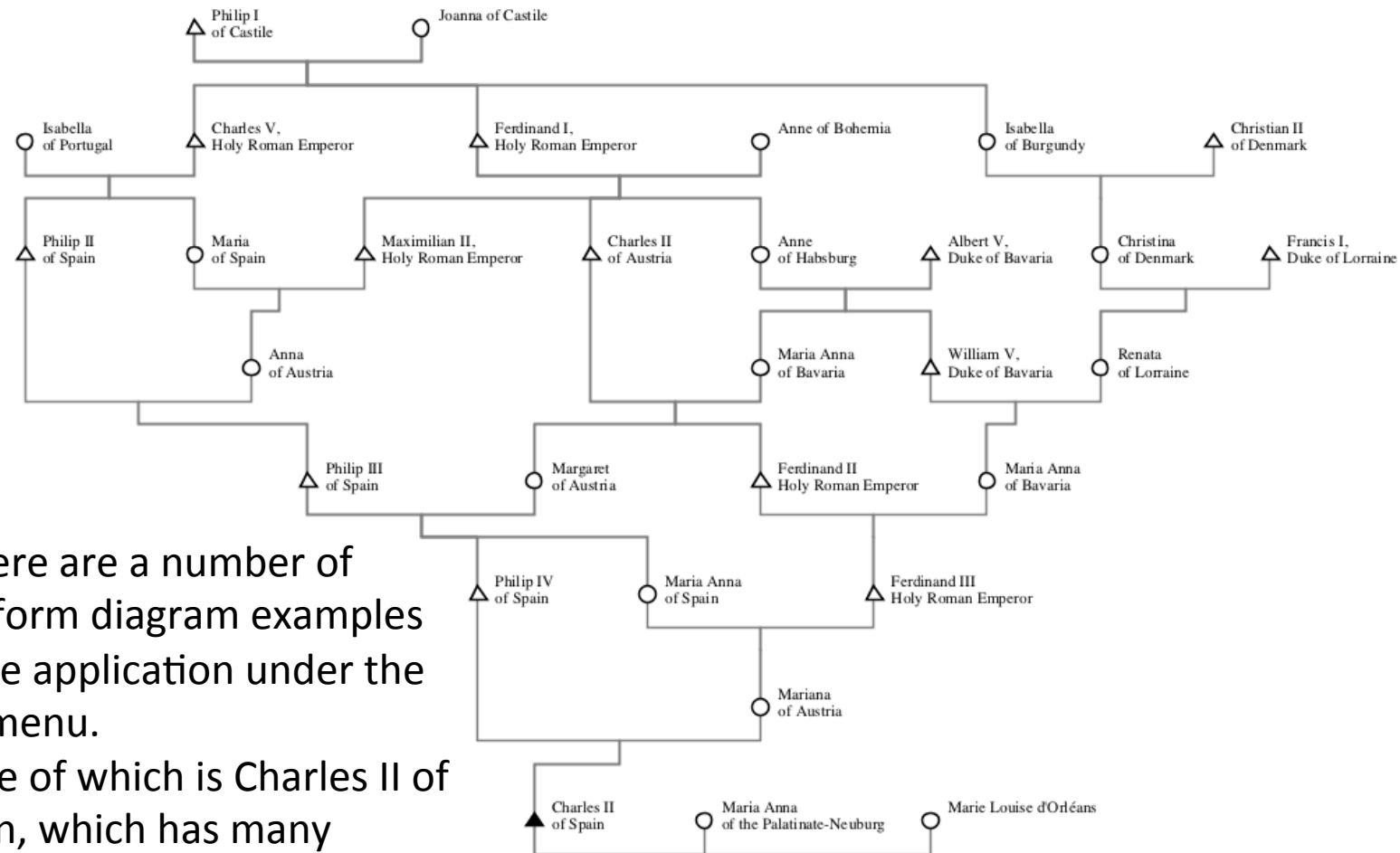
- A “Freeform Diagram” can be started from the file menu.
- Kin Type Strings are entered.
- Names and dates can also be added.



E ego, Em male ego, Ef female ego
M mother, F father, P parent
H husband, W wife
D daughter, S son, C child
Z sister, B brother



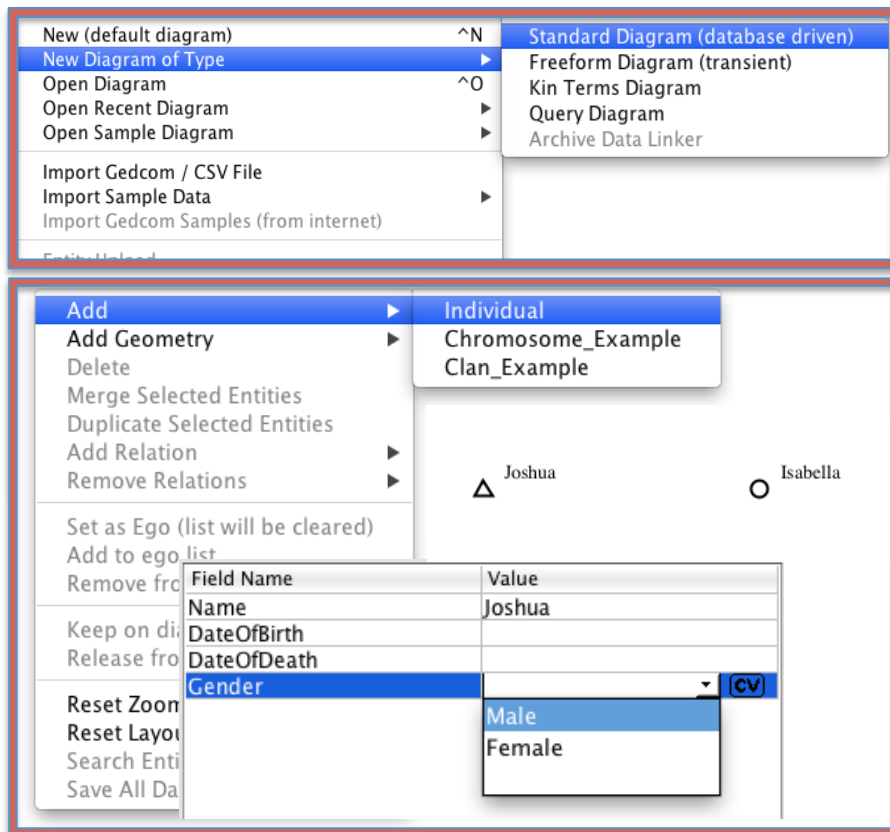
Freeform Example: Charles II of Spain



- There are a number of freeform diagram examples in the application under the file menu.
- One of which is Charles II of Spain, which has many matrimonial rings.

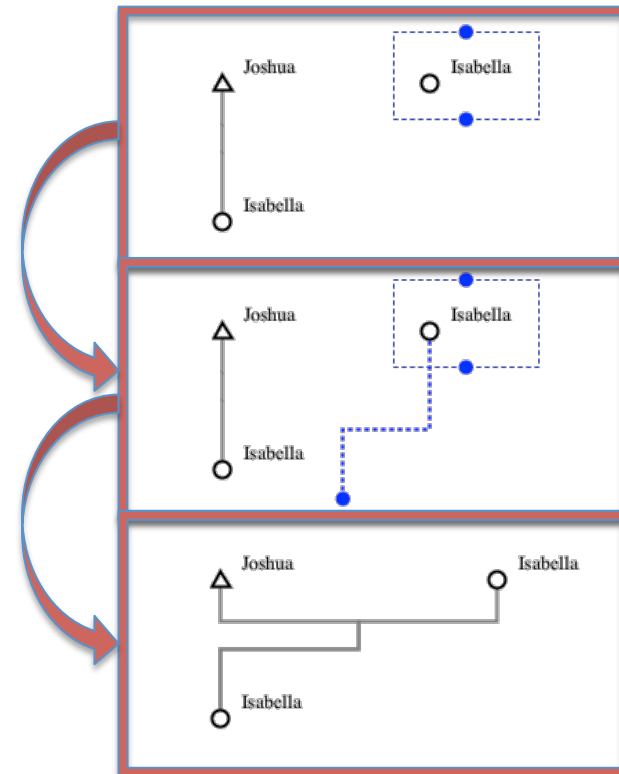
Project Diagrams

- A project can be opened or created via the file menu.
- Individuals/events/things can be added via the context menu.
- Relations can be added by dragging the blue dots on the selection.



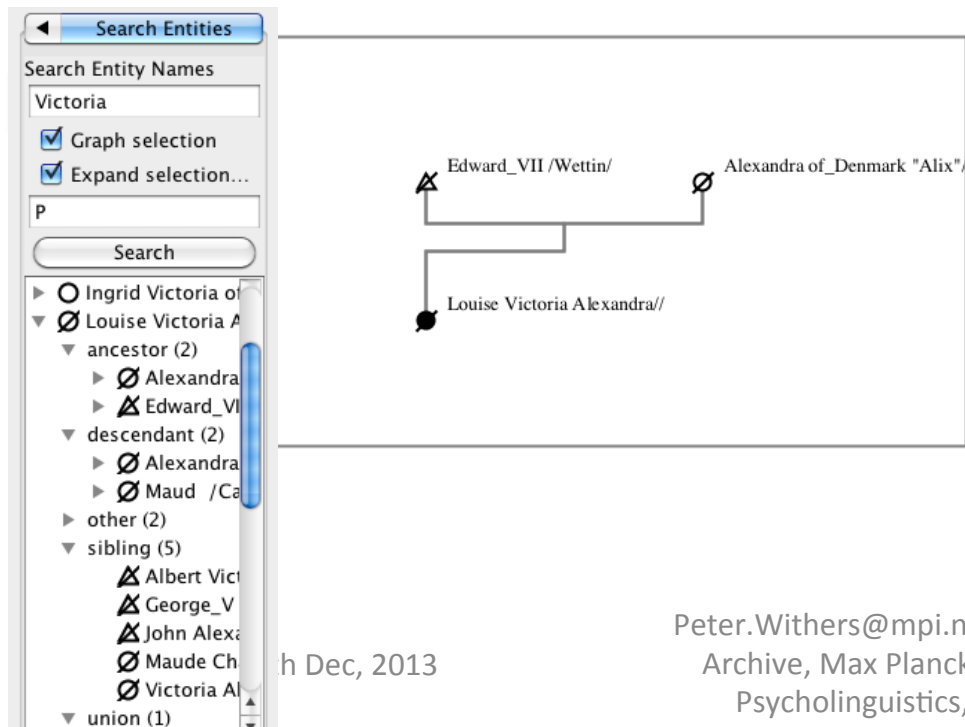
The screenshot shows two menu panels. The top panel is the 'File' menu, with 'New Diagram of Type' expanded to show options: Standard Diagram (database driven), Freeform Diagram (transient), Kin Terms Diagram, Query Diagram, and Archive Data Linker. The bottom panel is the 'Context' menu, with 'Add' expanded to show 'Individual', 'Chromosome_Example', and 'Clan_Example'. Below the menu is a diagram with two entities: Joshua (represented by a triangle) and Isabella (represented by a circle). A table below the diagram shows the 'Add' context menu options for the selected entity:

Field Name	Value
Name	Joshua
DateOfBirth	
DateOfDeath	
Gender	Male



Project Diagram: Search Tree

- Free text can be searched with fuzzy matching.
- The selected results can be inspected on the diagram.
- The selection can be expanded by kin type string.
- As the selection changes the diagram updates accordingly.
- The tree can also be browsed for immediate relations.



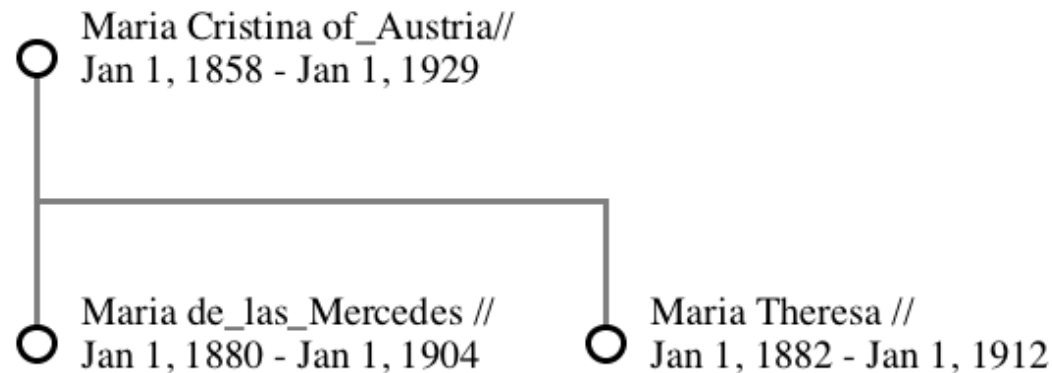
h Dec, 2013

Peter.Withers@mpi.nl The Language
Archive, Max Planck Institute for
Psycholinguistics, Nijmegen

Project Diagram: Kin Type Queries

- Kin type queries are used to retrieve individuals and their relations from the database.
- The kin types following a query will add any matching kin.
- For example the following query returns Maria and her daughters.

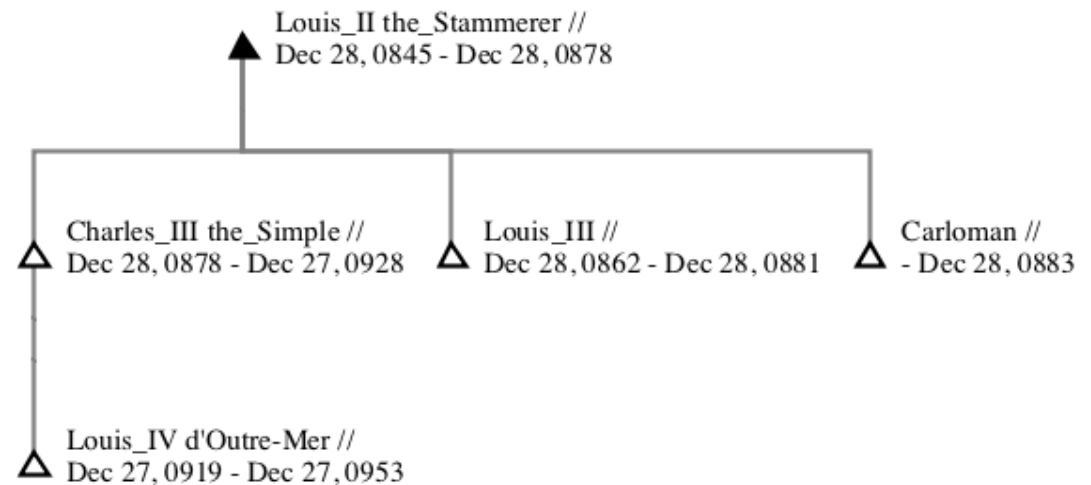
x[Maria Cristina of_Austria]D



Multiple Query Parameters

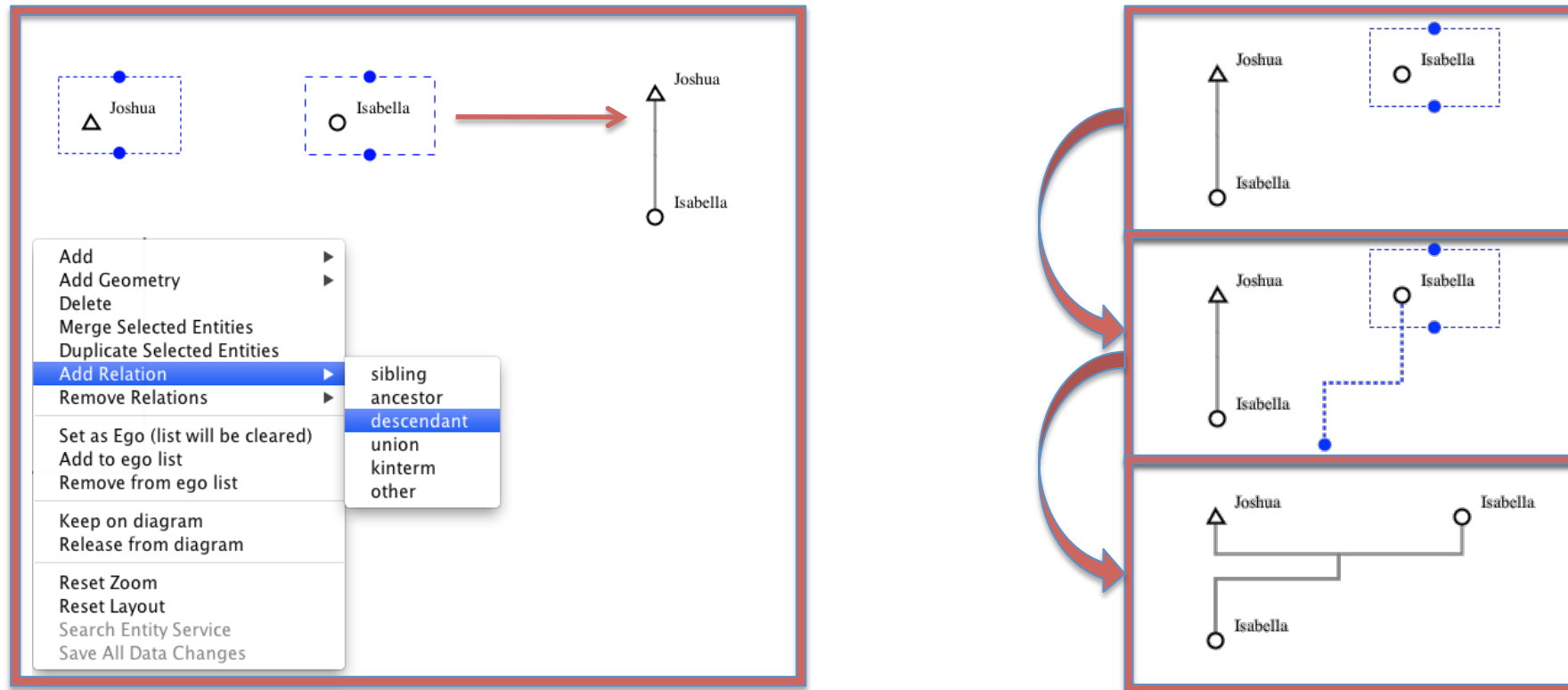
- Multiple queries can be used per kin type.
- Each condition can use:
 - = contains
 - == exact match
 - > greater than
 - < less than

`E[DateOfBirth<0850][INDI.TITL=King of France]CC`



Project Diagram: Creating Relations

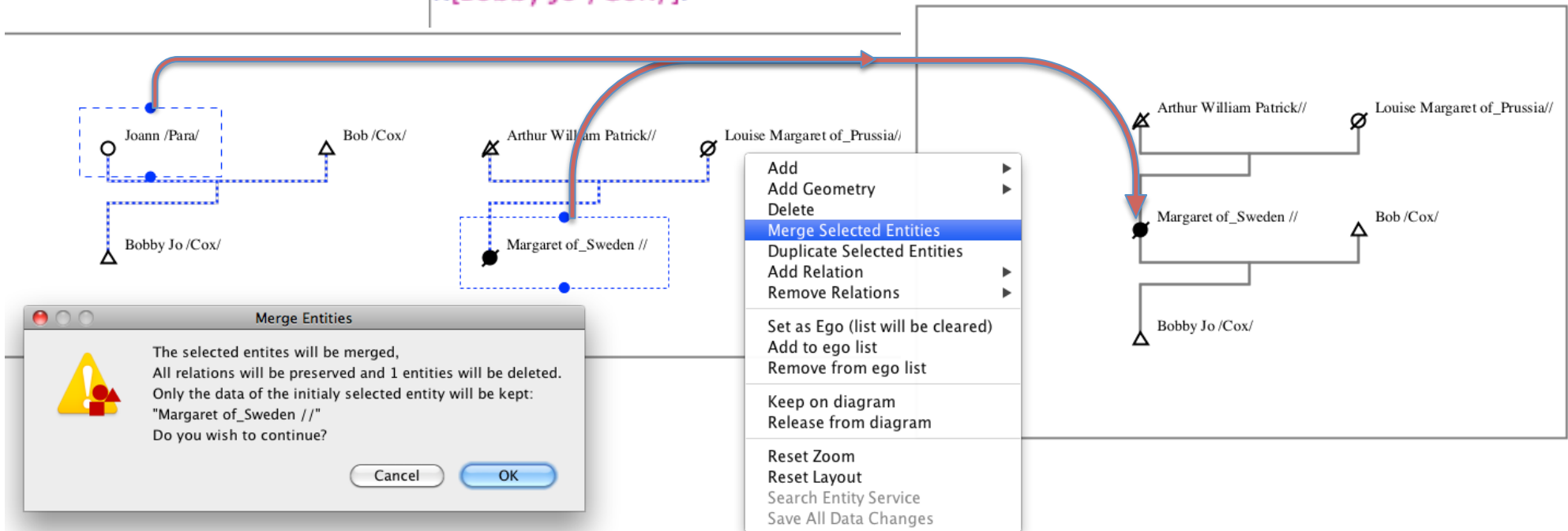
- Relations can be added via the context menu.
- Or via the drag handles of the selection.



Project Diagram: Merge Individuals

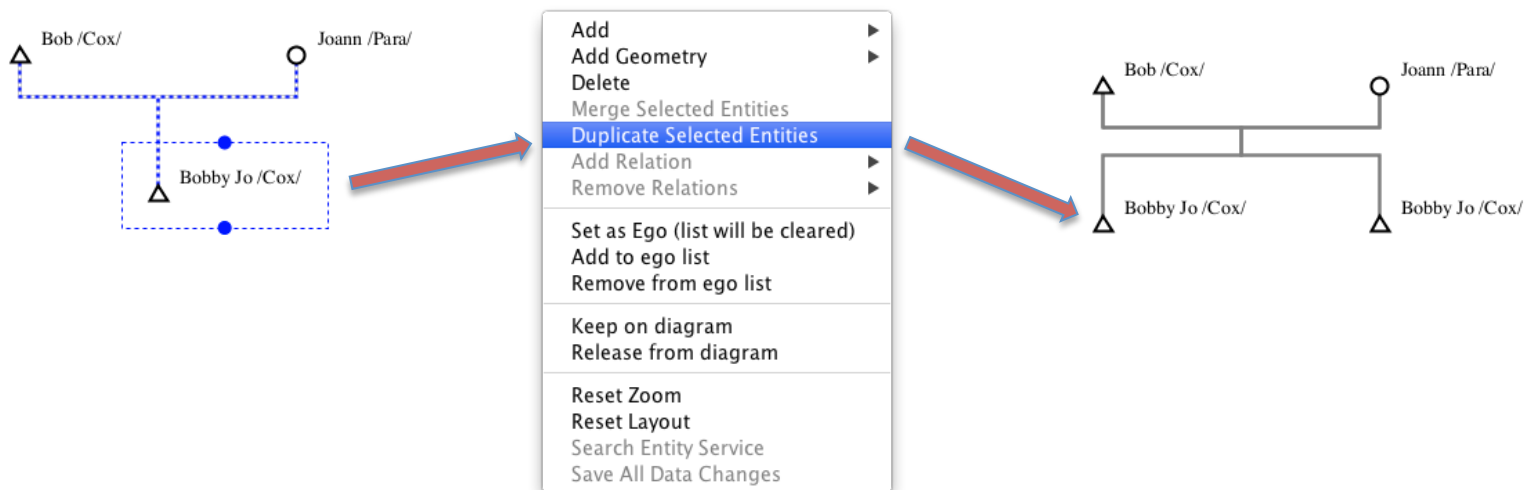
- When duplicate entries are found individuals can be merged, keeping the relations of both originals.

E[Margaret of_Sweden]P
x[Bobby Jo /Cox/]P



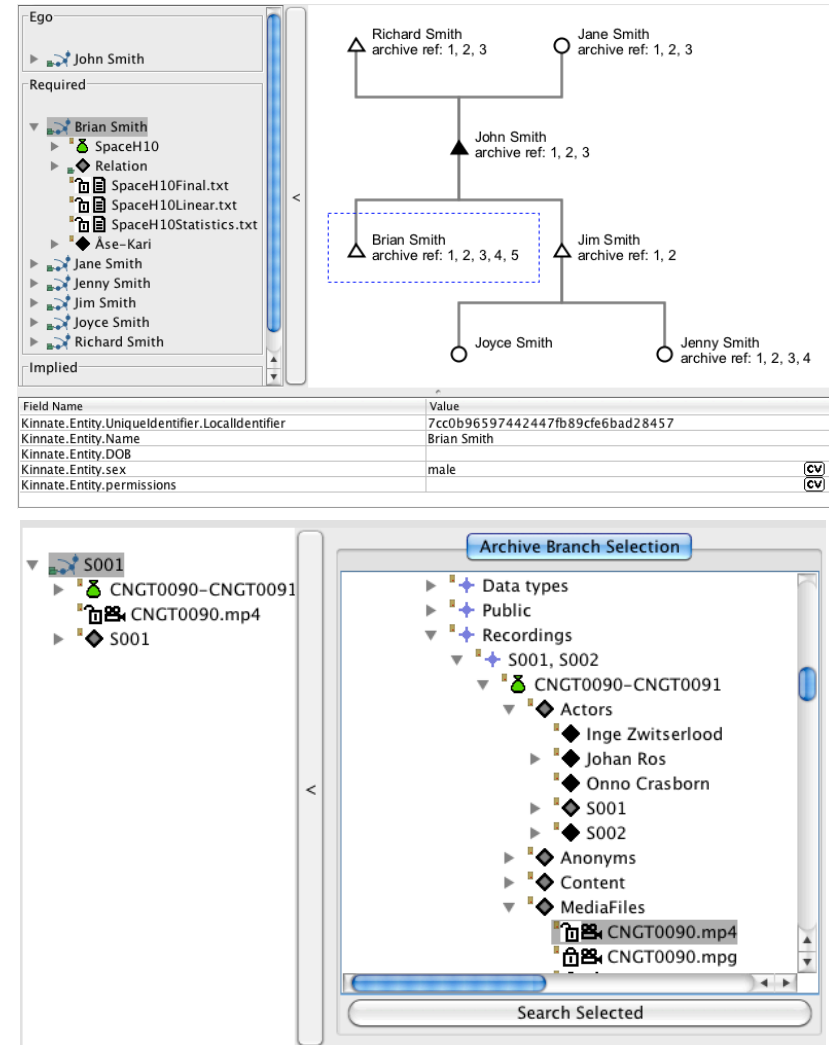
Project Diagram: Duplicating Individuals

- An individual can be duplicated.
- This will duplicate all relations.
- All kin data is also duplicated.



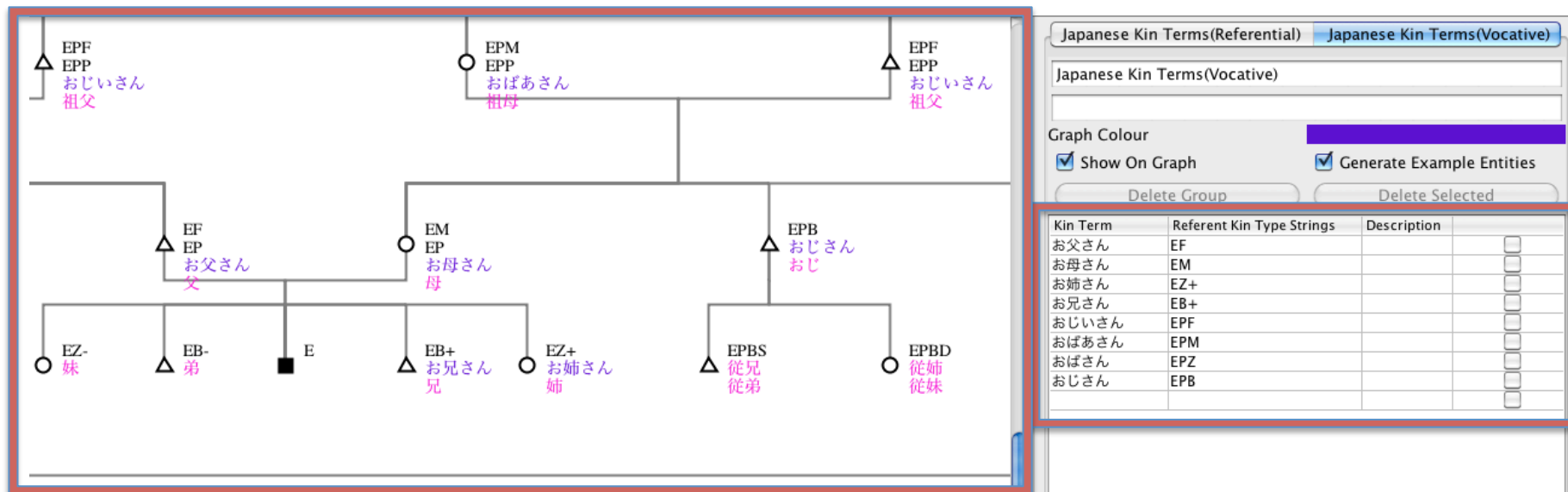
Project Diagram: Archive Linking

- Archive linking features will be extended in the next release.
- The current stable version has many of the linking features held back. This was because a technical agreement needed to be reached on how to persistently link kinship individuals to the archive metadata and data.
- It is also hoped that we will be allocated time to produce a light weight kinship panel for use in other applications, such as Arbil and ELAN.



Kin Term Diagrams

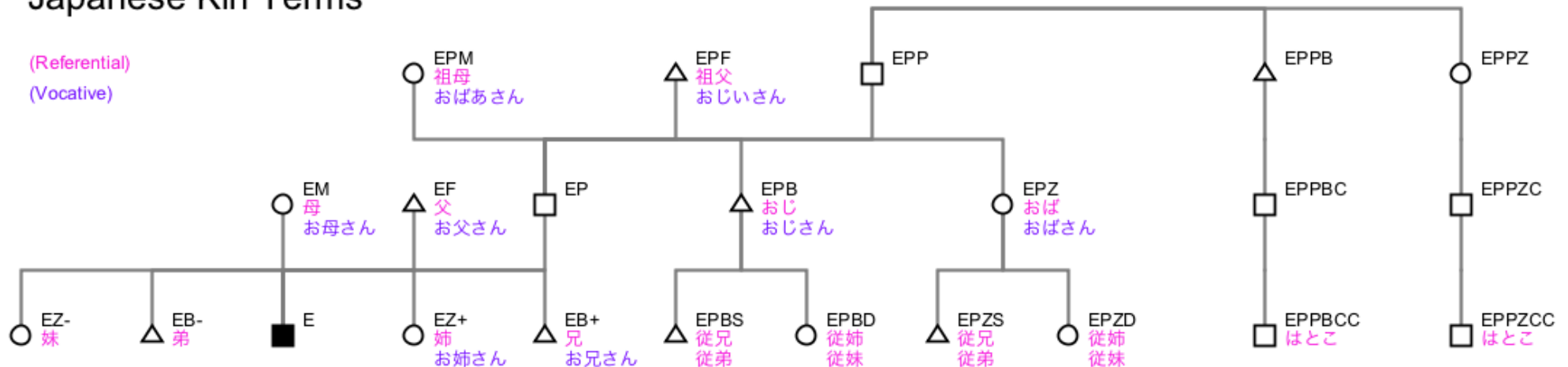
- Kin terms can be entered into a diagram.
- They are defined with kin type strings
- A freeform diagram can be generated from these kin terms.
- They can be imported and exported.
- In the future it will be possible to overlay these kin terms onto a database driven diagram.
- There are a number of example kin term diagrams in the application.



Comparative Kin Term Diagrams

- Multiple groups of kin terms can be shown on one diagram
- In this case some of the Japanese vocative and referential kin terms are shown on the diagram

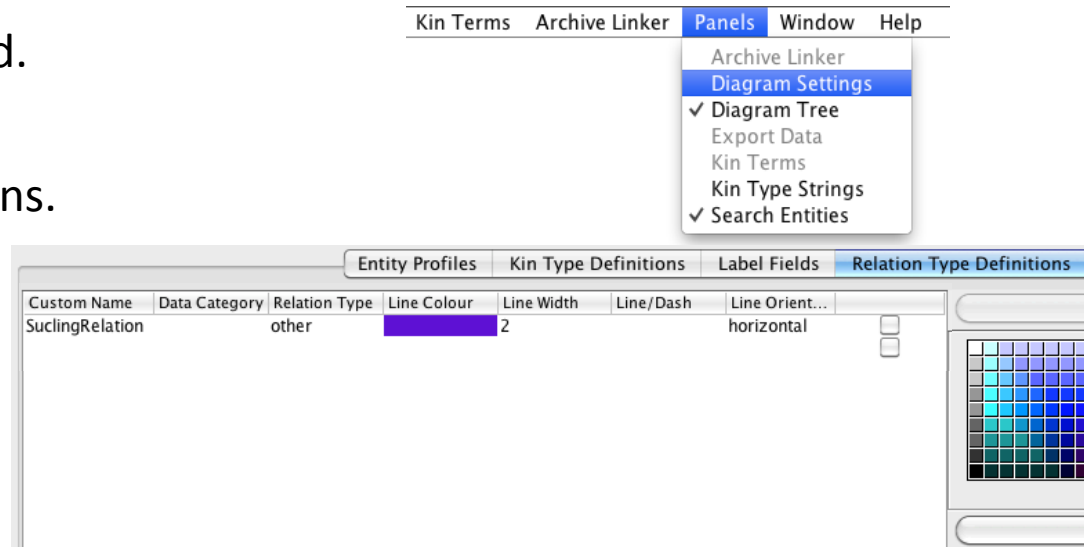
Japanese Kin Terms



Data sourced from <http://ja.wikipedia.org/wiki/親族> and subsequent links

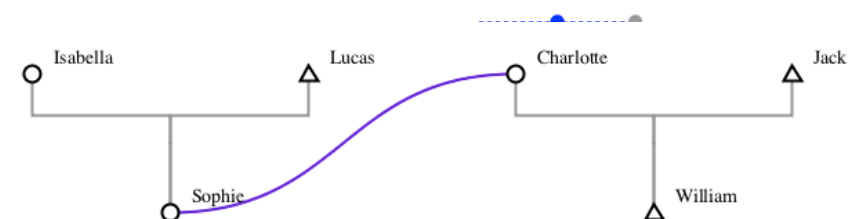
Custom Relation Types

- Custom relation types can be defined.
- This can be done in the diagram settings under relation type definitions.
- The custom name, type and display style can to be entered.
- The new relation type will then be available on the selected entities.
- These relations can be created by dragging the relation handles (dots).



The screenshot shows a software interface with a menu bar (Kin Terms, Archive Linker, Panels, Window, Help) and a 'Panels' dropdown menu. The 'Panels' menu is open, showing options: Archive Linker, Diagram Settings, ✓ Diagram Tree, Export Data, Kin Terms, Kin Type Strings, and ✓ Search Entities. Below the menu is a table titled 'Relation Type Definitions' with columns: Custom Name, Data Category, Relation Type, Line Colour, Line Width, Line/Dash, and Line Orient... The table contains one entry: 'SuckingRelation' with 'other' as the Data Category, a purple line color, and a width of 2. To the right of the table is a color selection palette.

Custom Name	Data Category	Relation Type	Line Colour	Line Width	Line/Dash	Line Orient...
SuckingRelation	other		purple	2		horizontal



The diagram shows a network of entities: Isabella, Sophie, Lucas, Charlotte, William, and Jack. Isabella and Sophie are connected by a vertical line. Lucas and Charlotte are connected by a vertical line. William and Jack are connected by a vertical line. A curved purple line connects Sophie and Charlotte, representing the 'SuckingRelation' type. Small circles (handles) are visible at the ends of the lines.

Custom Kin Data

- The kin data fields can be defined as a profile in the Clarin component registry.
- The **meaning** of each data field can be defined in the ISOcat data category registry.
- <http://catalog.clarin.eu/ds/ComponentRegistry>
- <http://www.isocat.org/>
- This means that you can specify the data fields that you need for your project.

Field Name	Value
Name	
DateOfBirth	
DateOfDeath	
Gender	CV
	Zachar
	Nekeveh
	"Intersex"
	Tumtum
	Aylonit
	Saris

Element: X

Name: Gender

ConceptLink:

Documentation:

DisplayPriority: 0

Min Occurrences: 1

Max Occurrences: 1

Type: Zachar

AttributeList:

- Zachar
- Nekeveh
- "Intersex"
- Tumtum
- Aylonit

Clarin Component Browser

Element: X

Name: DateOfDeath

ConceptLink:

Documentation:

DisplayPriority: 0

Min Occurrences: 1

Max Occurrences: 1

Type: [0-9]{4}/([0-9]{2})(0,2)(abt|bef|aft){0,1}

AttributeList: No Attributes

Element: X

Name: Gender

ConceptLink:

Documentation:

DisplayPriority: 0

Min Occurrences: 1

Max Occurrences: 1

Type: Male

AttributeList: No Attributes

Or create controlled vocabularies:

Item	appInfo	conceptLink
Male		
Female		
Click to add a row		

ISOcat
Data Category Registry
Defining widely accepted linguistic concepts

ISO 12620 provides a framework for defining data categories compliant with the ISO/IEC 11179 family of standards. According to this model, each data category is assigned a unique administrative identifier, together with information on the status or decision-making process associated with the data category. In addition, data category specifications in the DCR contain linguistic descriptions, such as data category definitions, statements of associated value domains, and examples. Data category specifications can be associated with a variety of data element names and with language-specific versions of definitions, names, value domains and other attributes.

[Access the Data Category Registry](#)

The Registration Authority of the TC 37 DCR is the [Max Planck Institute for Psycholinguistics](#), Nijmegen, The Netherlands.

This new DCR implementation is currently *under development* by the MPI in collaboration with TC 37. However, the core services should now be stable enough for day to day DCR usage. We welcome any *feedback* (bug reports, feature requests, etc.). You can use the following methods to contact us:

- file a [bug report or feature request](#)
- post a message on the [ISOcat forum](#)
- send an [email](#)

Kinship Data

Field Name	Value
Name	Isabella
DateOfBirth	
DateOfDeath	
Gender	CV
	Female
	Male
	Female

data from http://en.wikipedia.org/wiki/Third_gender#Israel

Custom Symbols

- A symbol can be any SVG definition.
- Custom symbols can only be manually inserted at this stage.
- A number of symbols are already included in the default diagram.
- These symbols can be associated with any kin data.
- The parameters for this can be entered manually or a data field can simply be dragged to the symbol definition list.
- Then a symbol can be selected from the diagram definitions.

The screenshot shows a software interface with three tabs: 'Label Fields', 'Relation Type Definitions', and 'Symbol Fields'. The 'Symbol Fields' tab is selected and contains a list of custom symbols, each with a dropdown menu and a small 'x' icon for deletion. The symbols listed are: triangle, circle, square, triangle, circle, triangle-90, square-45, triangle-180, and triangle-270. Below the list is a 'Diagram Tree' showing a hierarchy of symbols, and a 'Kinship Data' table with the following data:

Field Name	Value
Name	
DateOfBirth	
DateOfDeath	
Gender	Zachar (cv)

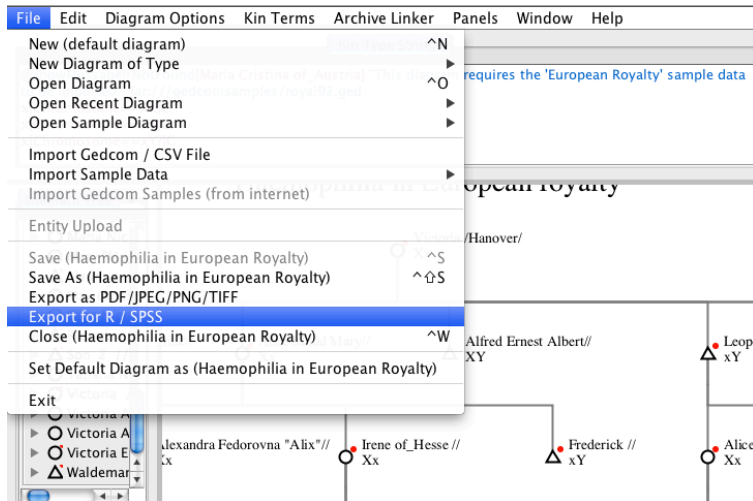
Gender data from http://en.wikipedia.org/wiki/Third_gender#Israel

R and SPSS

- The kinship data displayed on a standard diagram can be exported to a text file for use in R or SPSS etc.

```
dataFrame <- read.table("/home/peter/Desktop/KinOathDemoR/exportedData.tab",header=T)

# get all affected parent ids
affectedIds = dataFrame$id[dataFrame$symbol_redmarker == 1]
# get all children of affected
childrenOfAffected = dataFrame[dataFrame$momid %in% affectedIds || dataFrame$dadid %in% affectedIds]
# print names of affecteds' children
childrenOfAffected$label_0
```



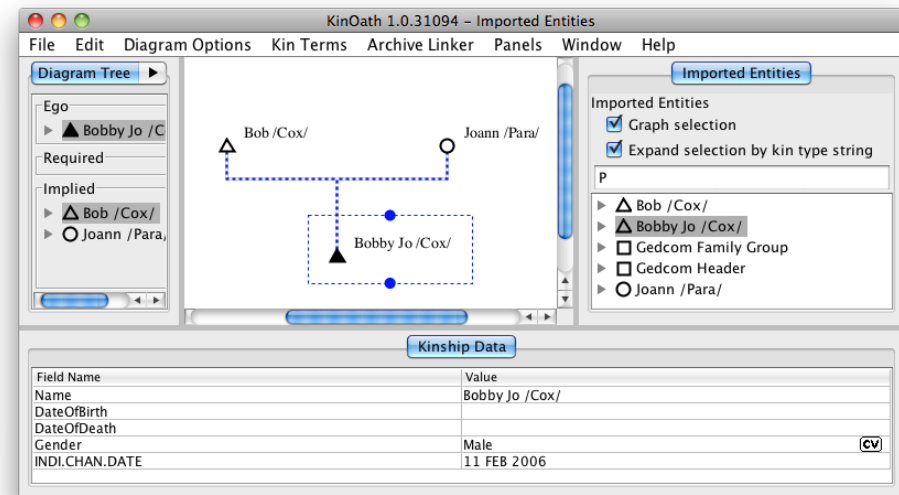
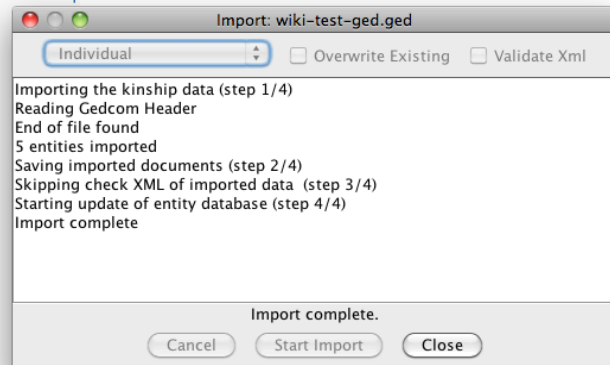
id	momid	dadid	sex	ego	dob	dod	label_0	symbol_triangle	symbol_blackstriketrough	symbol_redmarker	symbol_circle	UniqueIdentifier
1	21	42	1	0	1900	1904	Henry //	1	1	0	1	iid_1abe7dab1c70628f07963e95f9ebf3
2	37	43	2	0	1846/05/25	1923/06/09	Helena Augusta Victoria//	0	1	0	1	iid_ed611b28b31a0ec4c49801a0b67955c5
3	17	44	1	0			Son_2 //	1	0	1	0	iid_f165b998134ee9c87fcf777afcdcb30
4	21	42	1	0	1889	1945	Waldemar //	1	1	1	0	iid_01af3ba990cd7c5fb99c99663310fb5
5	37	43	2	0	1857/04/14	1944/10/26	Beatrice Mary Victoria//	0	1	1	0	iid_d573b6fabe6bc99b0dee7584a73f1c4c
6	29	45	2	0	1897/06	1918/07/18	Tatiana Nicholovna//	0	1	0	1	iid_61f27021af8823a0e385fa409ae578ad
7	14	46	1	0	1870	1873	Frederick //	1	1	1	0	iid_a8a378106e81d1a9358709ed2140da
8	32	47	1	0	1907	1938	Alphonso of_Cavadonga //	1	1	1	0	iid_7496663fda68b430682cf35d3e989f
9	37	43	1	0	1841/11/09	1910/05/06	Edward.VII /Wettin//	1	1	0	0	iid_15cfe4ec699efdf384eb3892d68164be4
10	37	43	1	0	1850/05/01	1942/01/16	Arthur William Patrick//	1	1	0	0	iid_55158ffe1e49fb5e03dccc7c9e7856b
11	37	43	2	0	1848/03/18	1939/12/03	Louise Caroline Alberta//	0	1	0	1	iid_04c5146f7a9f5602148fe498a2719272
12	32	47	2	0	1911		Maria Christina //	0	0	0	1	iid_38fe23e9e9198f80ec2bea370315527d
13	29	45	1	0	1904/08/12	1918/07/18	Alexis Nicolalevich /Romanov//	1	1	0	0	iid_9a0067ddf26ec47693a733a3c02694
14	37	43	2	0	1843/04/25	1878/12/14	Alice Maud Mary//	0	1	1	1	iid_67530193a6b8b2c0e1033446c23b55fe
15	14	46	2	0	1875	1878	Mary "May" //	0	1	0	1	iid_7b6fba6d2514f5ea6a2c0133f6481faf
16	5	48	1	0	1889	1922	Leopold //	1	1	1	0	iid_076ea3ded22a12f1dddf4bee9d9f5cd2
17	49	41	2	0	1883	1981/01	Alice of_Athlone //	0	1	1	1	iid_49c8d477a22646aa613021e236eb84
18	5	48	1	0	1886	1960/02/23	Alexander_of_Carisbrooke //	1	1	0	0	iid_7b59019bd3c76171a1ebaf6080225a
19	14	46	1	0	1868	1937	Ernest Louis of_Hesse//	1	1	0	0	iid_42b0e40a5a3ed02b29c62a3df8d29482
20	32	47	1	0	1914	1934	Gonzalo //	1	1	1	0	iid_d5699b660887d30bb18939c592c88d62
21	14	46	2	0	1866	1953	Irene of_Hesse //	0	1	1	1	iid_c7be01fab14a0b39a944cfc7b1af370
22	21	42	1	0	Child_#3 //			1	0	0	0	iid_2297e359f40a11761ca048204649f5a0
23	37	43	2	0	1840/11/21	1901/05/05	Victoria Adelaide Mary//	0	1	0	1	iid_cfb646cc3dff46971e79f0c3b0a853f
24	32	47	1	0	1913/06		Don Juan of_Spain//	1	0	0	0	iid_d0e1f42101710c32d6ae07940f3169e
25	29	45	2	0	1901/06	1918/07/18	Anastasia Nicholovna /Romanov//	0	1	0	1	iid_76a8b10d50c99d53a226f479ca8c3d0
26	37	43	1	0	1844/08/06	1900/07/30	Alfred Ernest Albert//	1	1	0	0	iid_icefe73a0656fef02c994ac13ec32e453
27	14	46	2	0	1864	1918/07/17	Elizabeth "Ella" //	0	1	0	1	iid_439671a86d8cf7ef7b64603d17c045c6
28	29	45	2	0	1899/05	1918/07/18	Maria Nicholovna /Romanov//	0	1	0	1	iid_ac0255ad17cb3b3f7ceef79a562c98fa
29	14	46	2	0	1872/06/06	1918/07/16	Alexandra Fedorovna "Alix"//	0	1	1	1	iid_2cf36745e277a718028035c9547243d8
30	5	48	1	0	1891	1914	Maurice //	1	1	0	0	iid_8719d4e8c7f5f343a017007e404c59
31	14	46	2	0	1863	1950	Victoria Alberta of_Hesse//	0	1	0	1	iid_7b00f979e251465c17888bf0d19a06aa
32	5	48	2	0	1887	1969	Victoria Eugenie "Ena"//	0	1	1	1	iid_e6f08317358fc7605bba50b0947c5fa1
33	32	47	1	0	1908	1975	James //	1	1	0	0	iid_f278191b0089852d4c52275213190fb
34	32	47	2	0	1909		Beatrice //	0	0	0	1	iid_92724995a7618cc9cf9c8d177b0b5a
35	29	45	2	0	1895/11	1918/07/18	Olga Nicholovna /Romanov//	0	1	0	1	iid_f236828d50909413df16564a9c44ba10
36	17	44	2	0	1906		May Cambridge //	0	0	0	1	iid_5e66bb2dab0a3bf454a255deec69a
37	50	51	2	0	1819/05/24	1901/01/22	Victoria /Hanover//	0	1	1	1	iid_87b8650d6336994f9f24654fdb214c71
38	49	41	1	0	1884	1954	Charles Edward //	1	1	0	0	iid_7efe2ede2bd8d488c46bc7d6f8b1179
39	32	47	1	0			Don Jamie //	1	0	0	0	iid_3dede17c537c1aa874ba9b0ad4972
40	17	44	1	0	1928		Rupert //	1	1	1	0	iid_8b561f64886573a2b5164f687d4d02a4
41	37	43	1	0	1853/04/07	1884/03/28	Leopold George Duncan//	1	1	1	0	iid_72fd2eb4dde4353125894161fc8dcd

Gedcom Import

sample.ged

```
0 HEAD
1 SOUR Reunion
2 VERS V8.0
2 CORP Leister Productions
1 DEST Reunion
1 DATE 11 FEB 2006
1 FILE test
1 GEDC
2 VERS 5.5
1 CHAR MACINTOSH
0 @I1@ INDI
1 NAME Bob /Cox/
1 SEX M
1 FAMS @F1@
1 CHAN
2 DATE 11 FEB 2006
0 @I2@ INDI
1 NAME Joann /Para/
1 SEX F
1 FAMS @F1@
1 CHAN
2 DATE 11 FEB 2006
0 @I3@ INDI
1 NAME Bobby Jo /Cox/
1 SEX M
1 FAMC @F1@
1 CHAN
2 DATE 11 FEB 2006
0 @F1@ FAM
1 HUSB @I1@
1 WIFE @I2@
1 MARR
1 CHIL @I3@
0 TRLR
```

- GEDCOM is the most common kinship data format.
- These files can be imported via the file menu.
- All GEDCOM fields are imported.
- Resource links are preserved.
- For testing the “GEDCOM 5.5 Torture Test Files” are used: <http://www.geditcom.com/gedcom.html>



Example from: <http://en.wikipedia.org/wiki/Gedcom>

Plugin Structure

- A plugin framework has been developed and is already available in KinOath and in Arbil
- This allows the stable versions to be more of a thin application, which is easier to test and maintain.
- New features can added as plugins.
- Plugins can be developed by third parties.
- This will allow for example:
 - External projects to create components, such as PUCK developed at EHESS.
 - Alternative data sources could also be supported as plugins.
 - A greater variety of graph sorting algorithms could be created, such as sort by clan vs descendancy etc.
 - Specialised export plugins will be possible.
- Another benefit of this structure is that new features do not need to have such an impact on the core applications stability.

Changes in 1.4

- The table now has an add/delete field menu, this allows free text entry for field names and a list of known field names to insert.
- Scrolling the diagram up/down/left/right via the mouse wheel now moves the diagram in a way more consistent with that used in other applications.

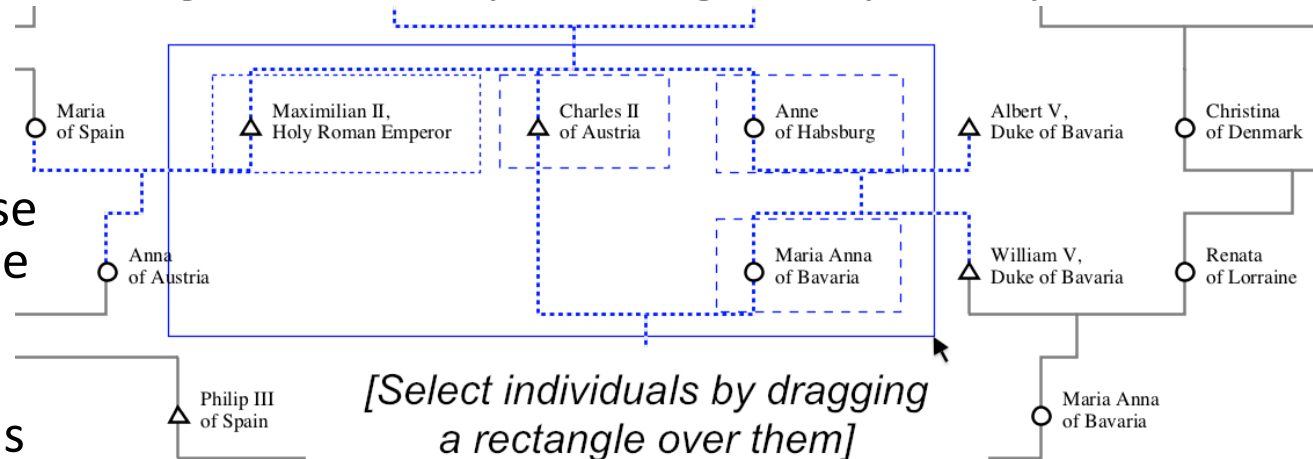
Field Name	Value
Name	老子
DateOfBirth	6th century BCE
DateOfDeath	
Gender	

Nickname
Notes
SOUR
SurnameInherited
SurnameMarried

Add
Add (T)
Add <custom field>
Delete Field "Gender"

Changes in 1.4

- Individuals on the diagram can now be group-selected with a rectangular selection box by dragging the mouse over them.
- Corrected resizing behaviour for graphics elements when dragging via the blue drag handle dots.
- The diagram settings for labels and symbols now allow the user to enter more complex settings while still providing a simple way to enter the settings.
- Editing a field in the table and other such actions no longer cause the current zoom to be lost.
- In addition, selecting an individual now pans to that individual without changing the zoom on the diagram.

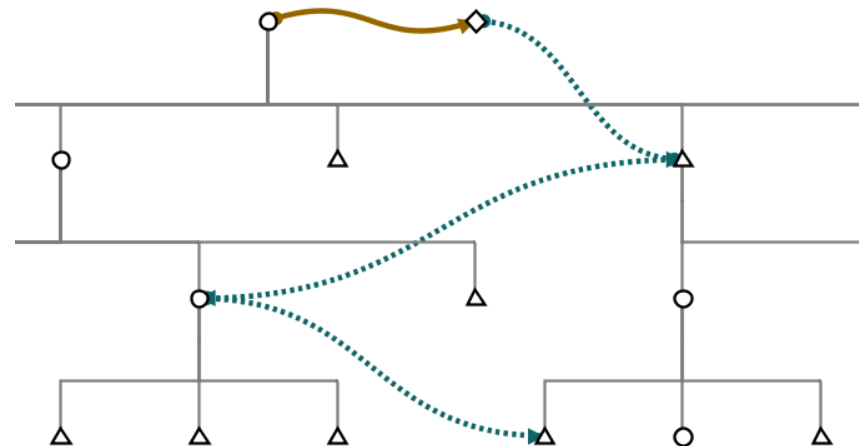


Changes in 1.4

- Adding a relation no longer changes the position of the related individuals/entities.
- When adding/removing relations, the application now responds faster (more improvements in this area will be made soon).
- Directed relations are now available. These are relations that are unidirectional and have an arrow to indicate the direction. These can be used, for instance, to record kin term use between individuals.
- The diagram progress bar is now shown when adding relations to individuals and when deleting, merging, and duplicating entities in the diagram.
- Refreshing list of available profiles in the diagram settings is now quicker.
- Entities without a label and without an icon are now more visible in the project trees and in the diagram.

Custom Name	Relation Type	Line Colour	Line Width	Line/Dash	Curve Line Orientation
SongAuthorship	directedout (directedin)	Orange	4	0	horizontal
SongInheritance	directedout (directedin)	Green	4	3	horizontal

Song Inheritance Example



Changes in 1.4

- When importing GEDCOM files, redundant FAM entities are discarded after importing the relations, unless they have useful data such as a marriage date etc.
- The search can now span separate data fields, such as first name field / last name field. It also now supports fuzzy matching so that similar entries or misspellings can be found.
- The project data can now be exported to a single file via a new export plugin. The format is similar to GEDCOM but without the cultural restrictions. (This will export all fields and all relations. However, concepts that are not allowed in the GEDCOM specification will create non standard GEDCOM file output.)
- Support for custom relation types has been improved and they are now easier to configure. For instance setting the line colour and style is now easier. The custom relations settings also now clearly indicate when there is an opposing relation type, such as ancestor/descendant.

Starting KinOath Kinship Archiver

- Open the MPI TLA tools page:
<http://tla.mpi.nl/tools/tla-tools/kinoath>
- The webstart version is usually easiest
- Alternatively choose the installer for your operating system



- Note that Java is required for all versions

Conclusion

- KinOath desktop is available for download and the latest version addresses issues raised in user feedback.
- The current version has many features, such as:
 - Create publishable-quality kinship diagrams;
 - Create kin term diagrams with multiple kin term groups or language groups on the diagram;
 - Create, view, import, export kinship data.
 - Create, view, import, export kin terms data
 - Query large datasets based on free text and kin type strings.
 - Quickly create complex or simple diagrams.
 - Customise the kin data, kin terms, kin types, relation types and symbols to suit your needs.
- This application is financially accessible to everyone (free) .