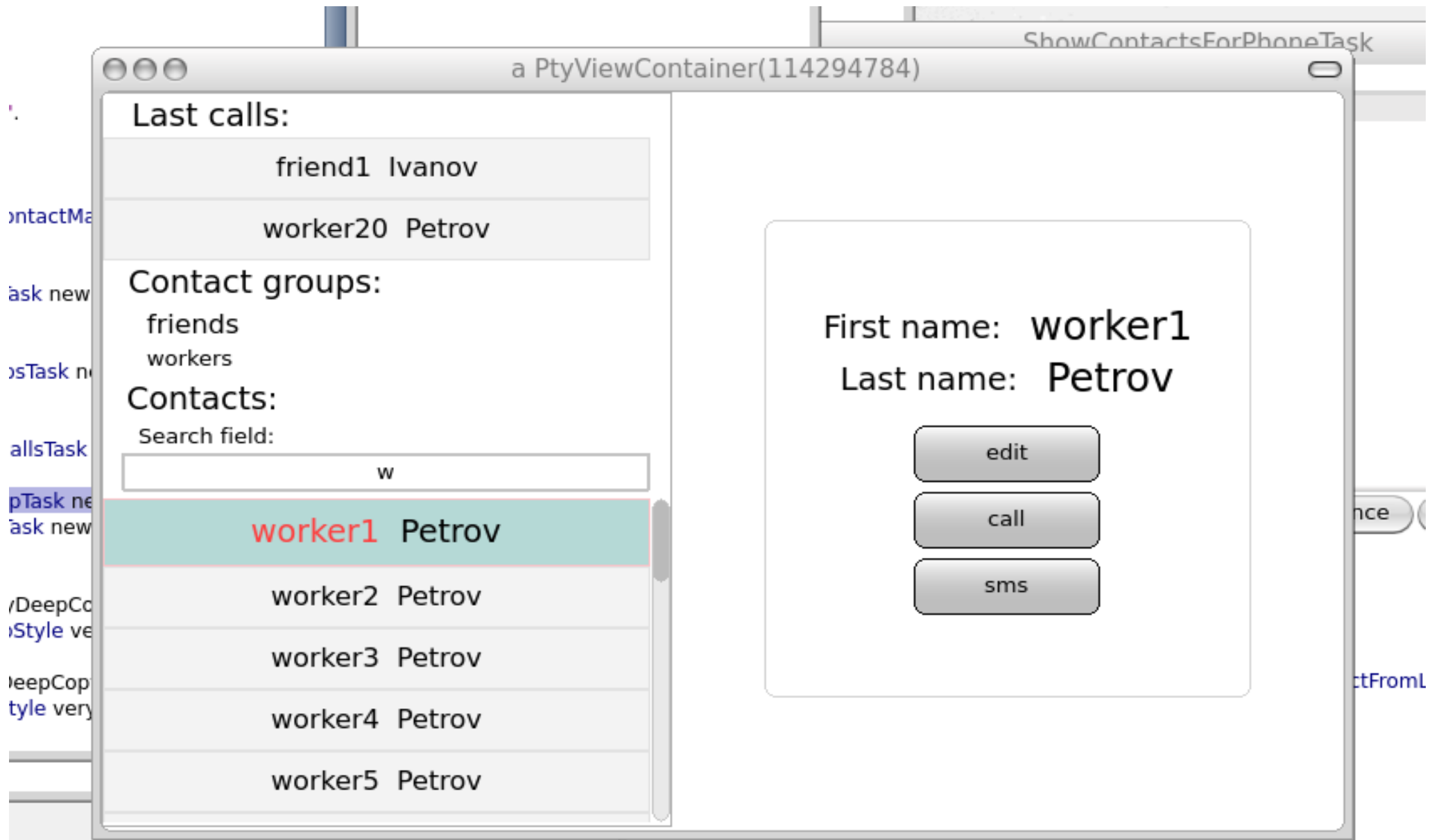


Presenty User Interface Framework

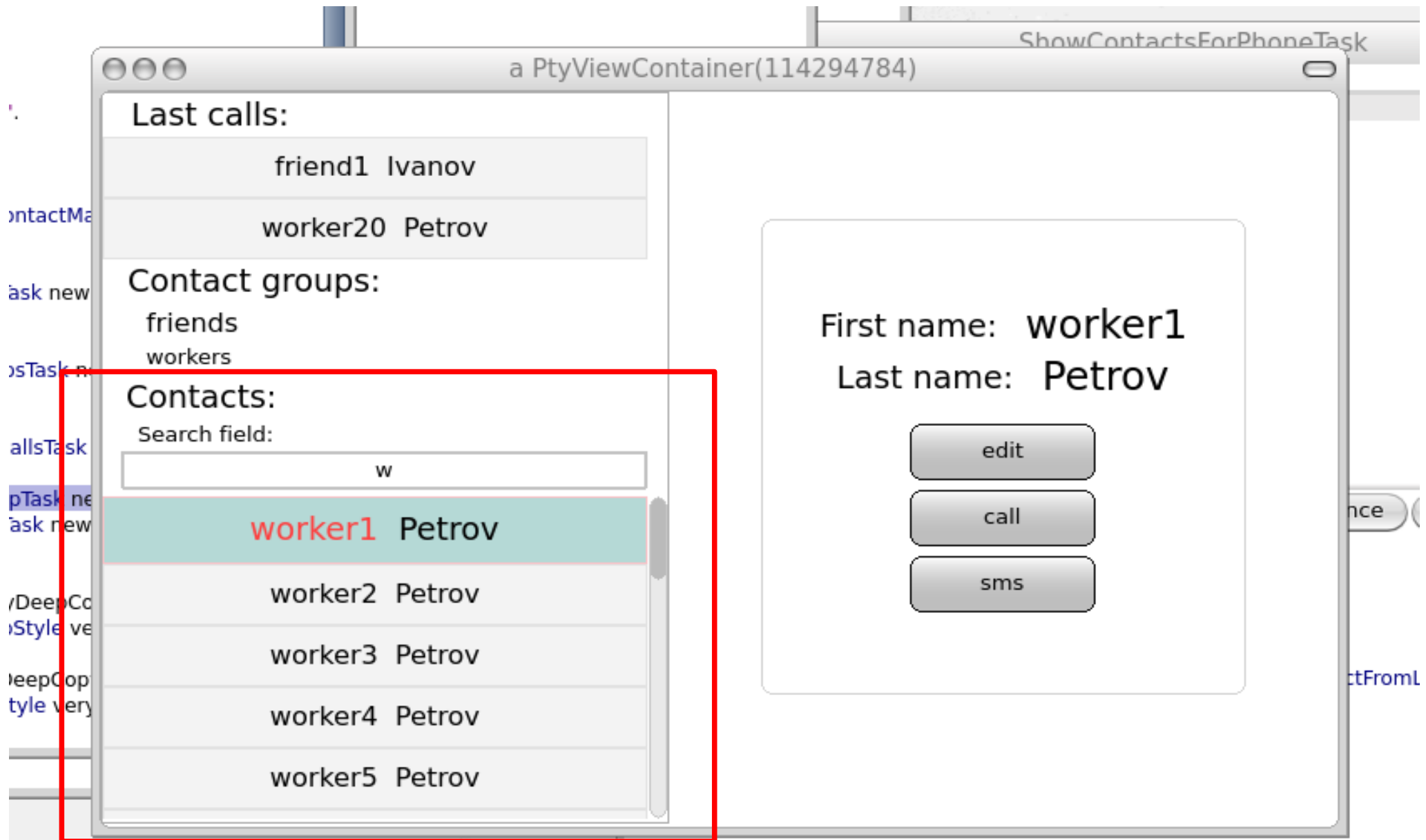
Presenty

Platform-independent user interface
application

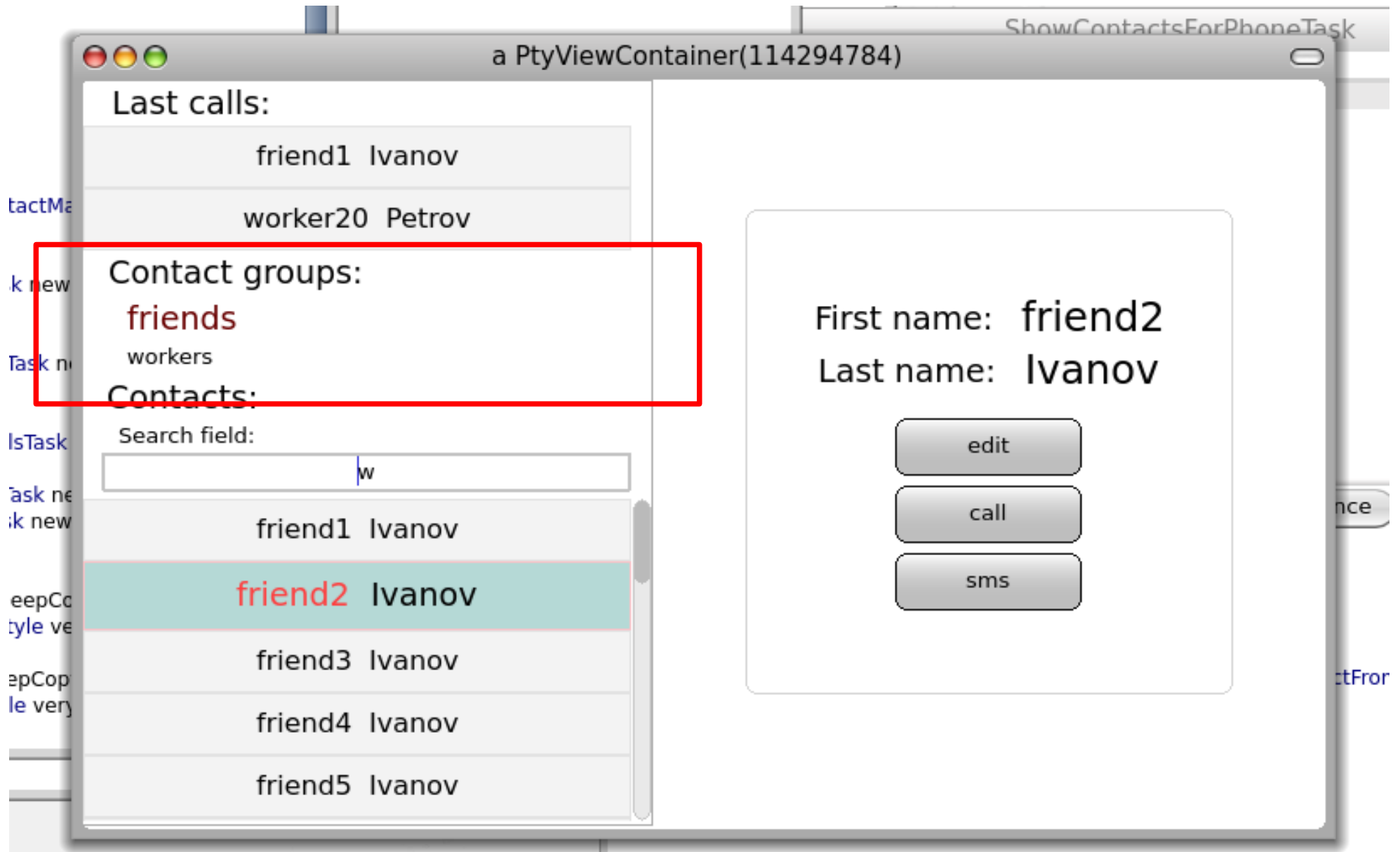
Desktop contact manager



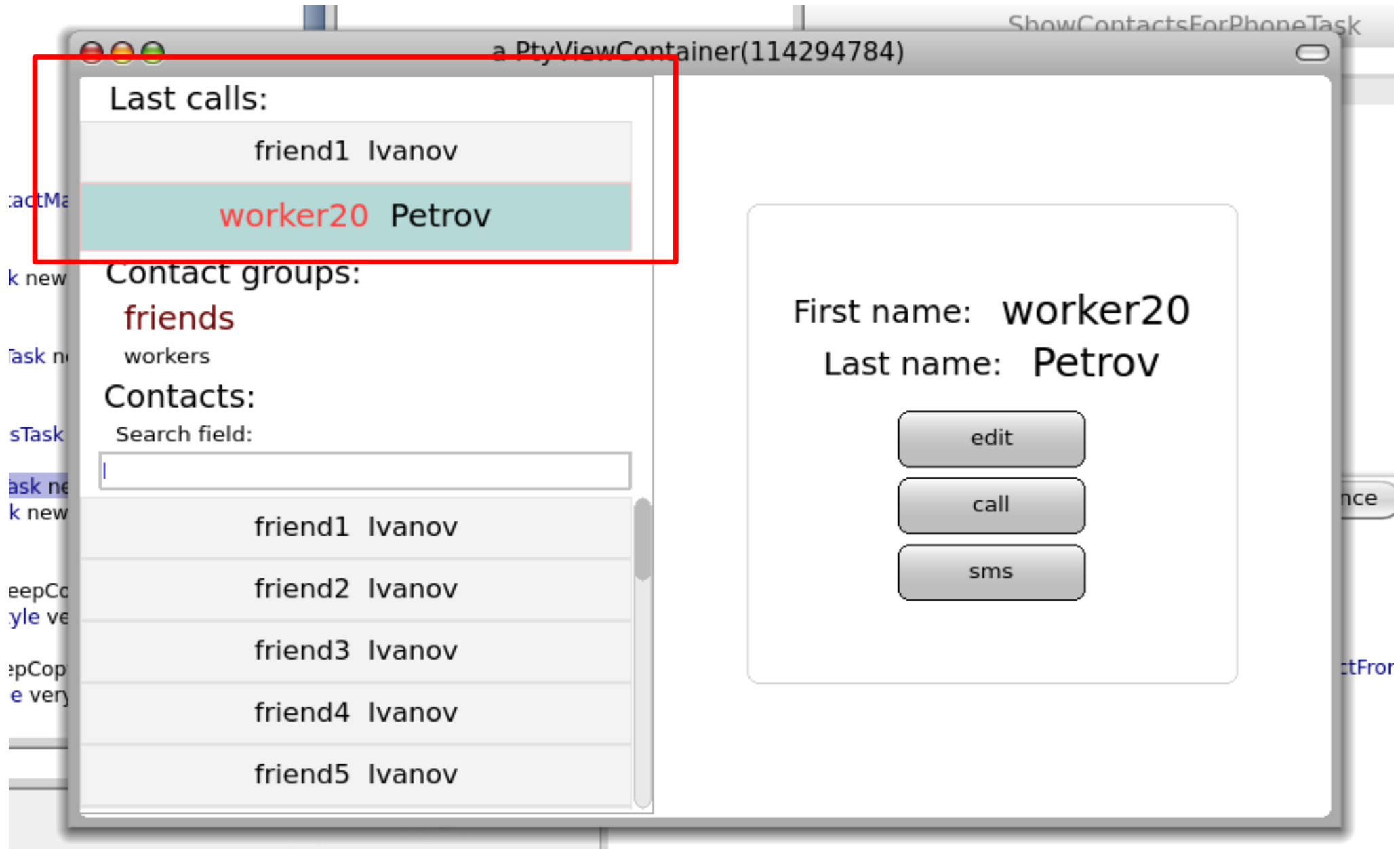
Filter contacts by typed name



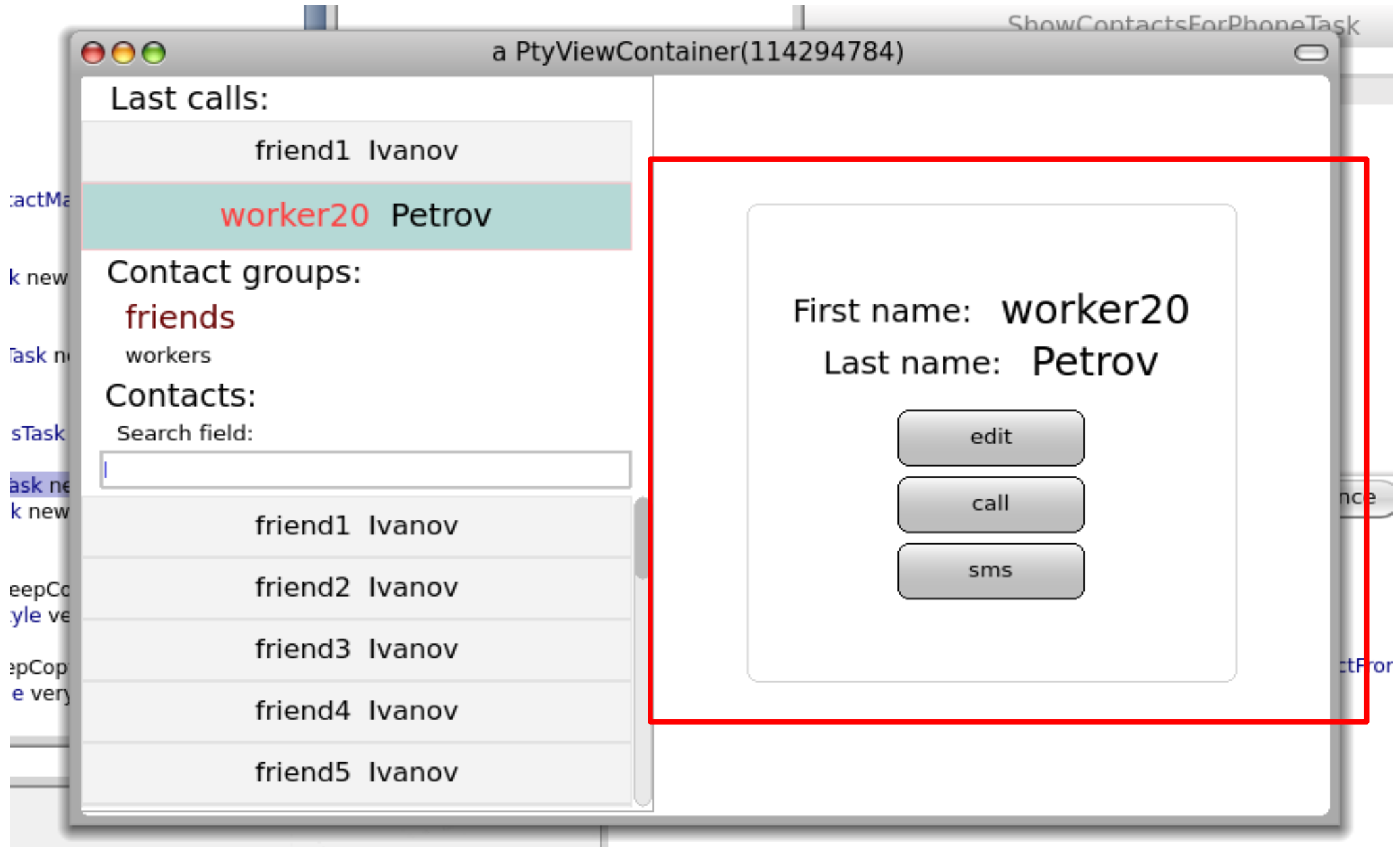
Select contact from groups



Select contact from last calls



Contact selection show details data



Presenty approach

- Multiple separated tasks a user can perform
- Each task is a sequence of user requests
 - Contact groups selection task:
 - 1) select group from groups list
 - 2) select contact from selected group contents
 - 3) look at selected contact details

Implementation

- Continuation-based tasks as elements of user interaction at application level
- User requests as building blocks for tasks:
 - Primitive user requests to invoke base interface components
 - Compound requests for calling other tasks
- Extendable DSL (pure Smalltalk) facilitating task implementation
- Modified Model-View-Presenter pattern

SelectContactFromGroupsTask

body

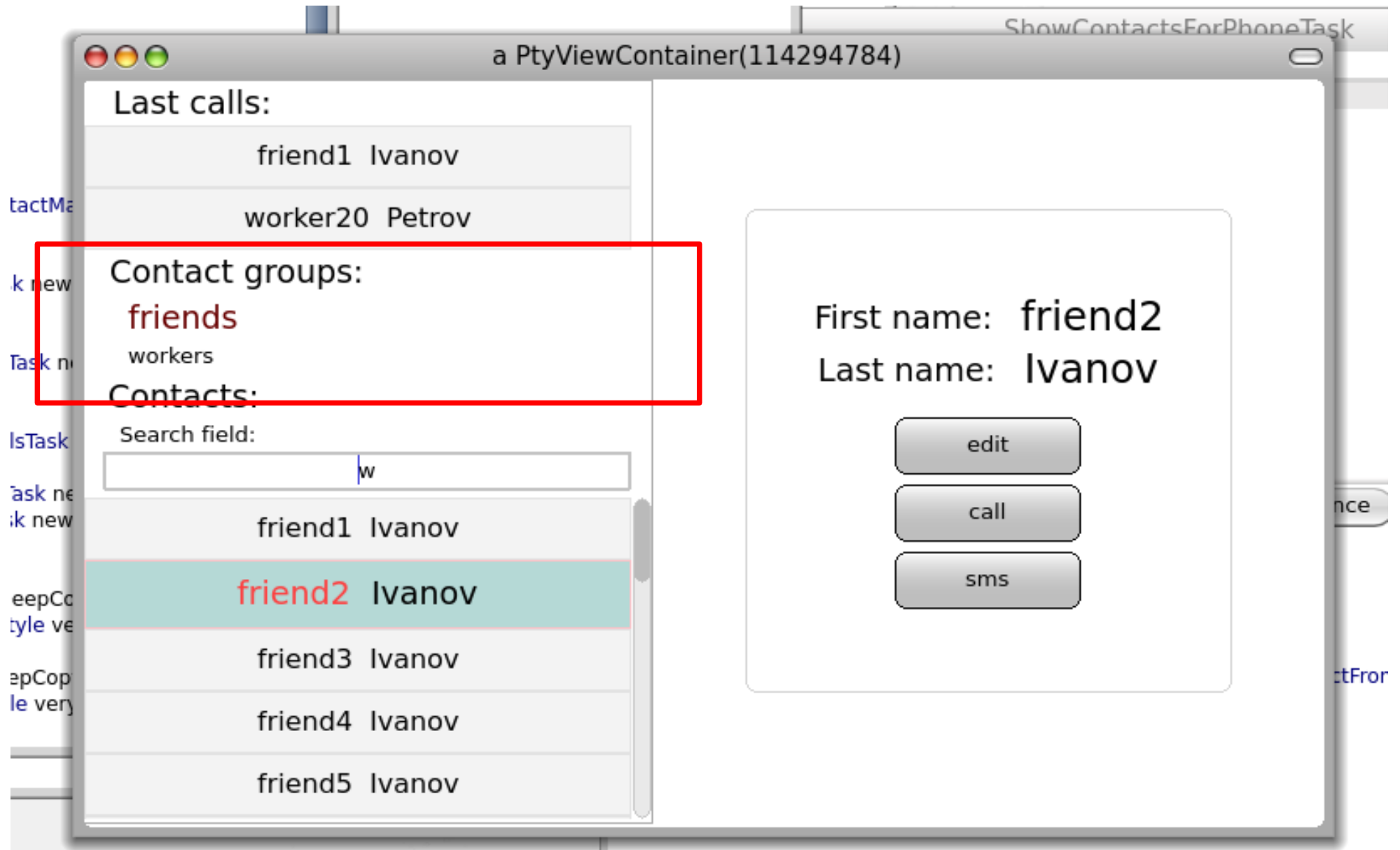
| selected group |

group := user select: ContactsGroup from: ContactManager groups.

selected := user select: Contact from: group contents.

user goTo: (DetailedContactPresenter on: selected)

Select contact from groups



SelectContactBySearchTask

body

| selected |

selected := user execute: (

 PtySearchItemTask forItems: ContactManager contacts ofClass: Contact).

guide restoreActiveArea.

user goTo: (DetailedContactPresenter on: selected)

SelectContactFromLastCallsTask

body

| selected |

selected := user select: 'LastContact' from: ContactManager lastCalls.

user goTo: (DetailedContactPresenter on: selected)

ShowContactsForDesktopTask

body

guide viewport createNewViewArea.

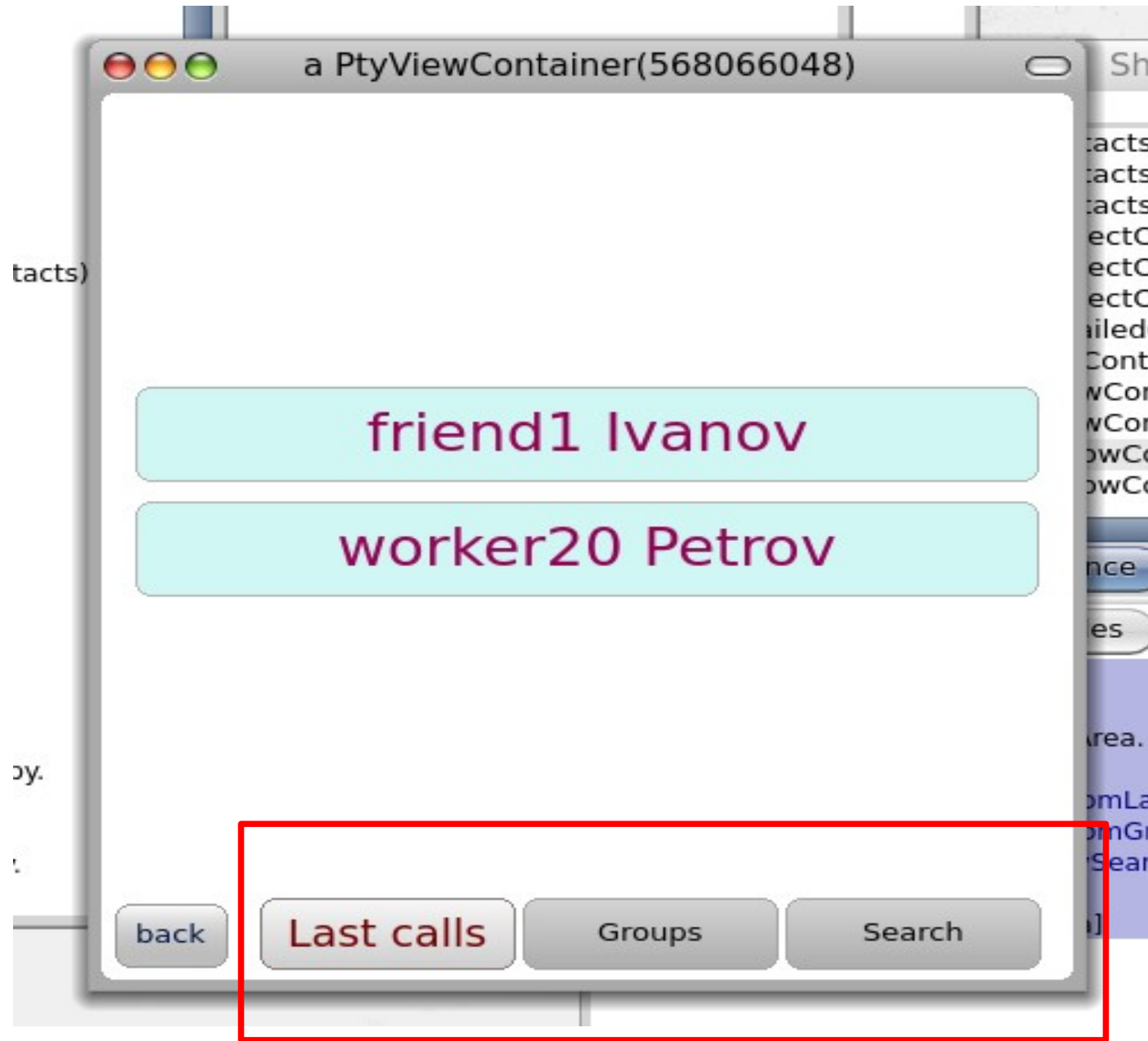
guide forkTask: SelectContactFromLastCallsTask new.

guide forkTask: SelectContactFromGroupsTask new.

guide forkTask: SelectContactBySearchTask new.

user goTo: [guide showViewArea]

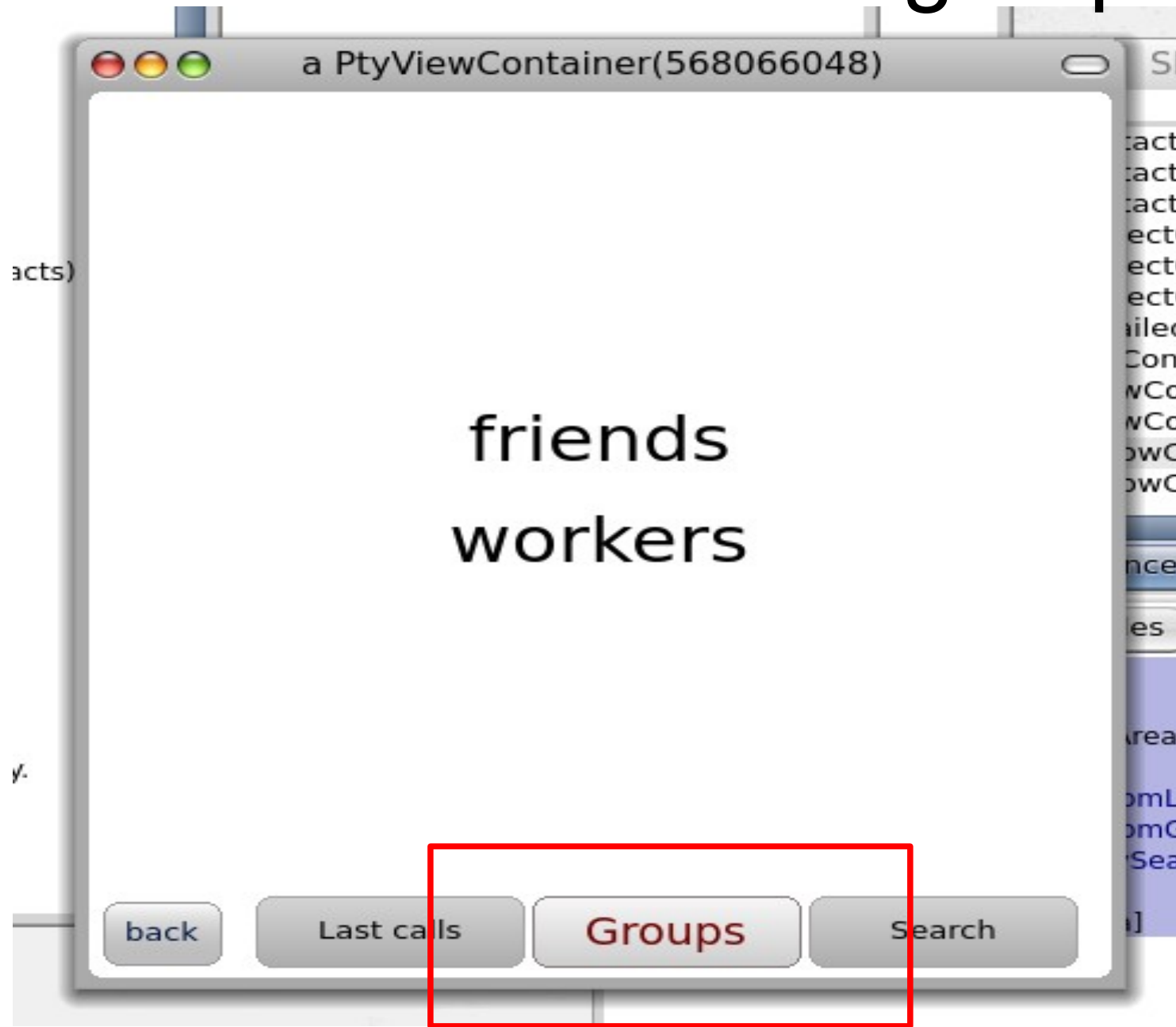
Select contact from last calls



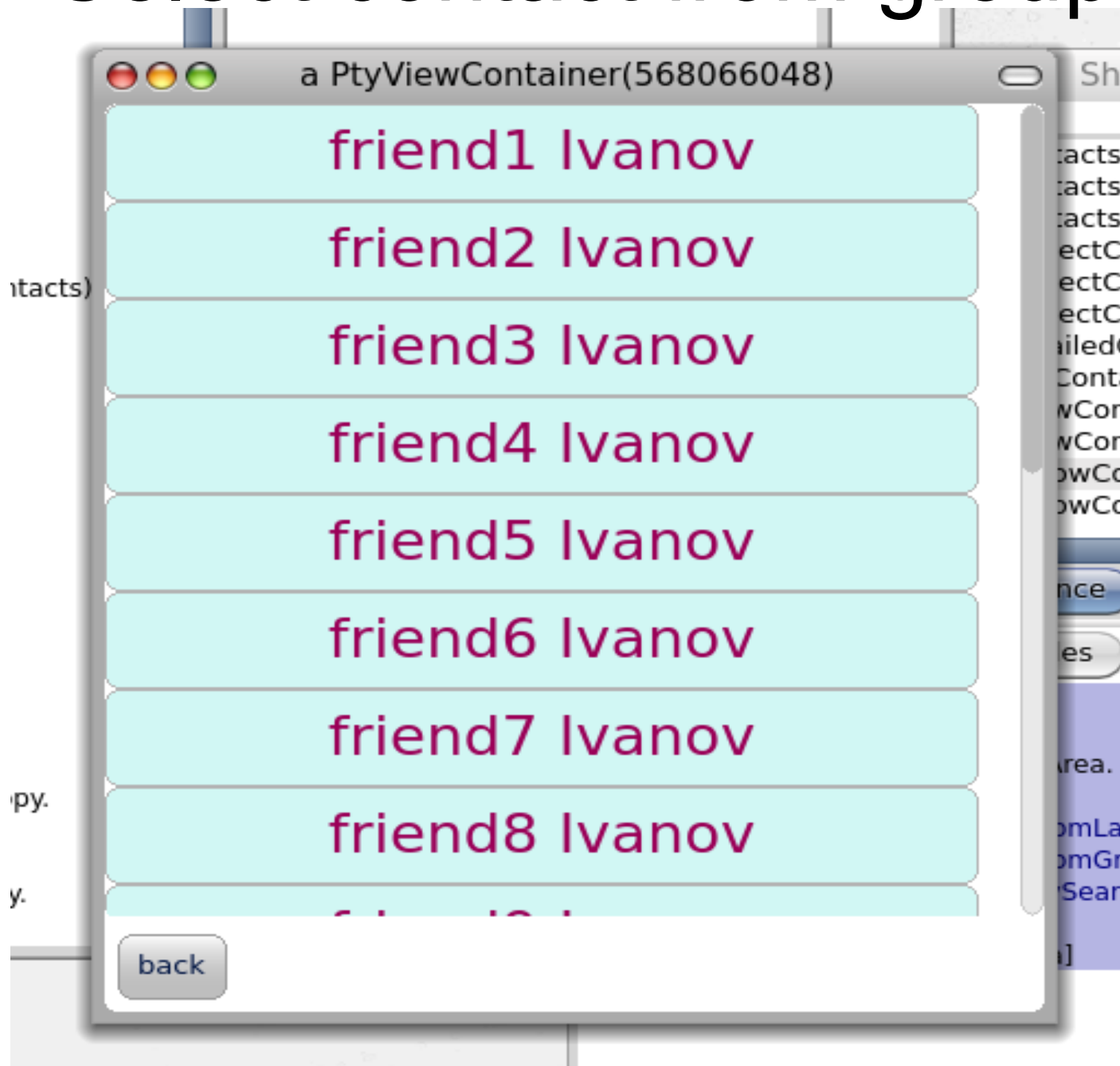
Select contact from last calls



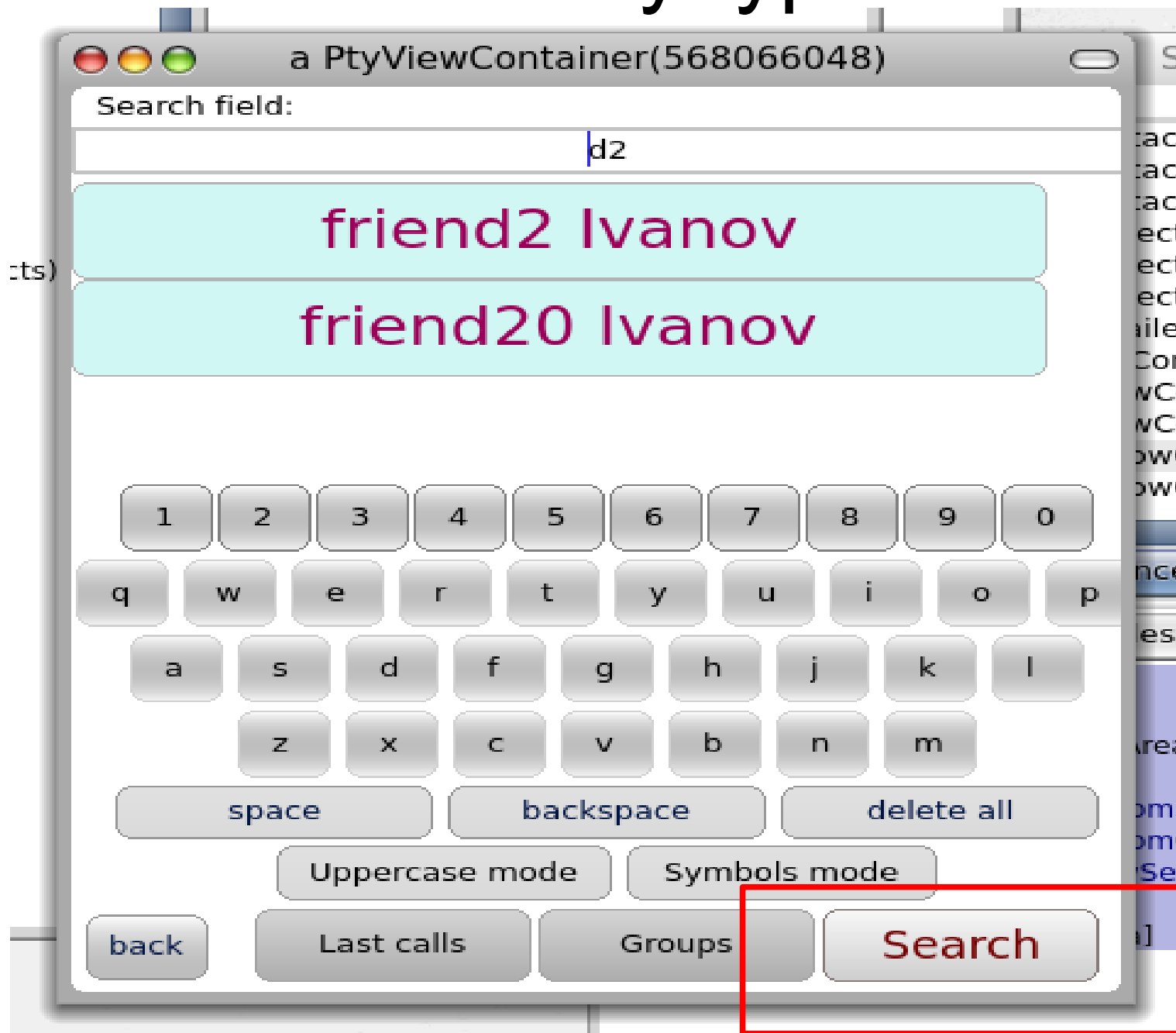
Select contact from groups



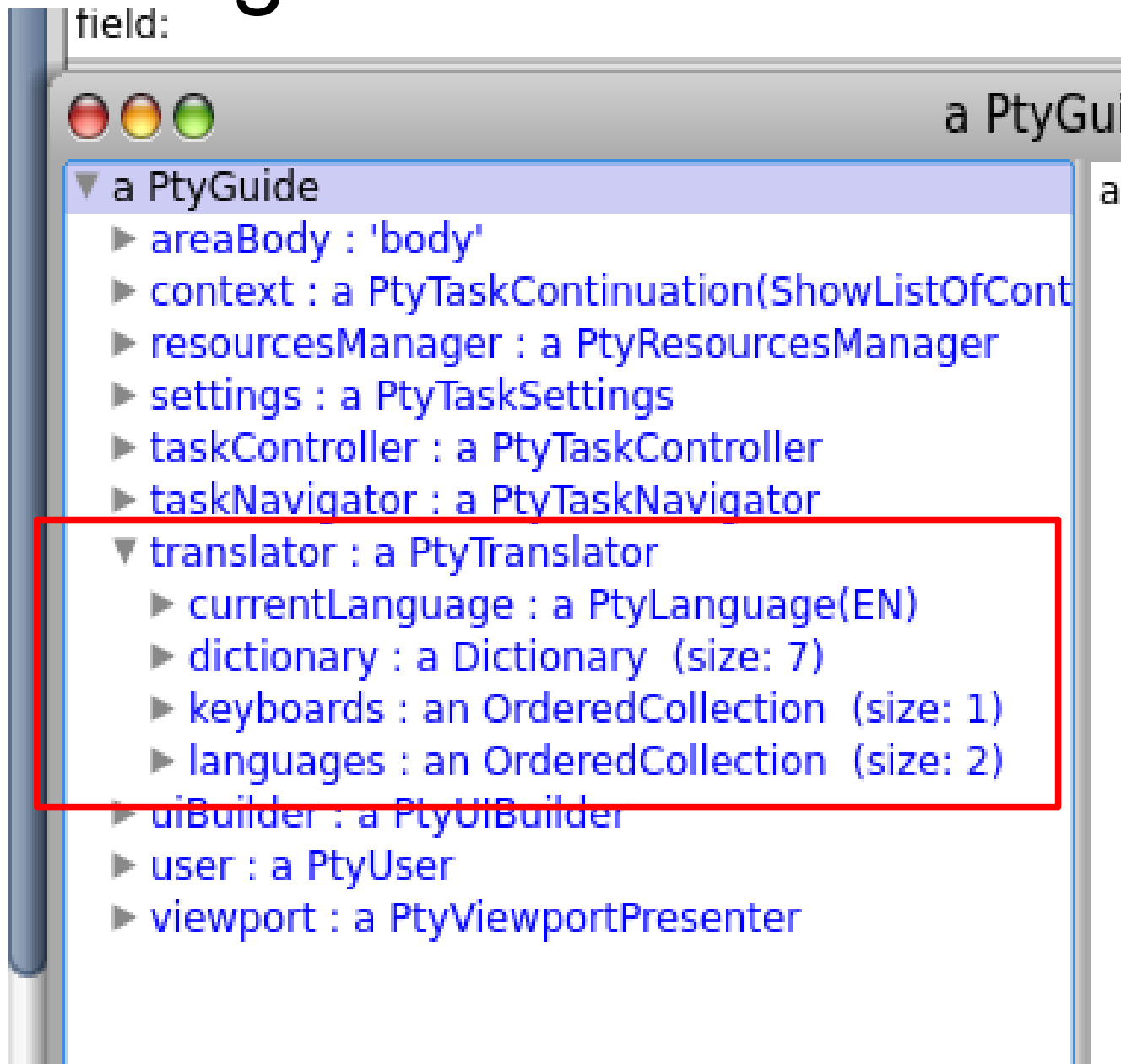
Select contact from groups



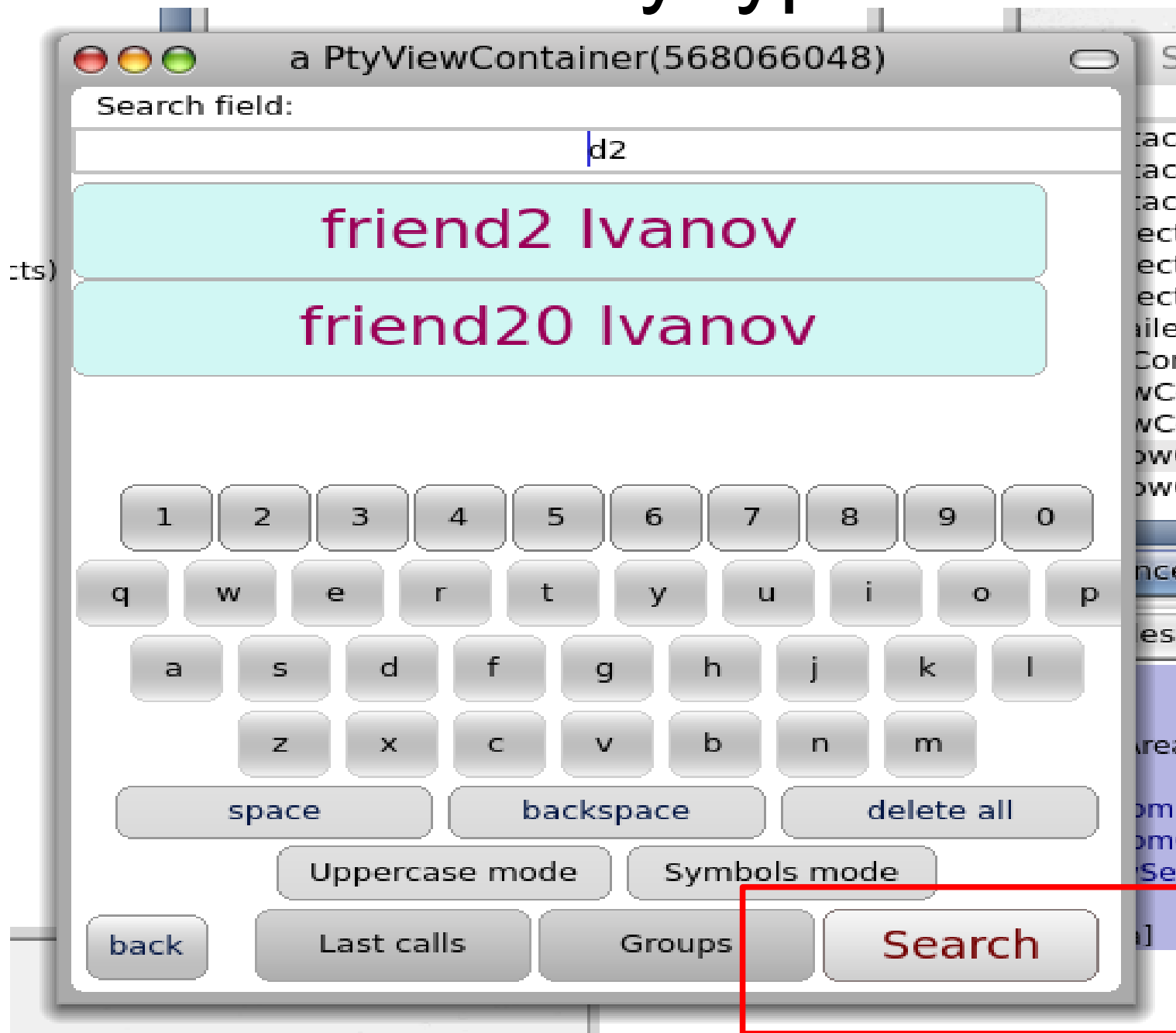
Filter contacts by typed name



guide translator



Filter contacts by typed name



Filter contacts by typed name



ShowContactsForPhoneTask

body

| taskClass |

guide viewport createNewViewArea.

taskClass := user select: 'ContactManagerTask' from:
{SelectContactFromLastCallsTask. SelectContactFromGroupsTask.
SelectContactBySearchTask}.

guide addBackButton.

user execute: taskClass new

Task Configuration

- Presenter can have different views in different contexts
 - For different parent presenters
 - For different tasks
 - For different styles of presenter
 - For different styles of parent presenter

Task Configuration

- Guide can store default settings for tasks
 - guide create: PtyShowListTask
- Task settings can vary in different contexts

How tasks configured

- PtyItemsListTask has itemPresenterFactory which can be saved in settings
 - PtyExplicitPresenterFactory specified concrete presenter class to present list items
 - PtyPresenterPreviewFactory use #preferredPreviewPresenter message for getting presenter from list item

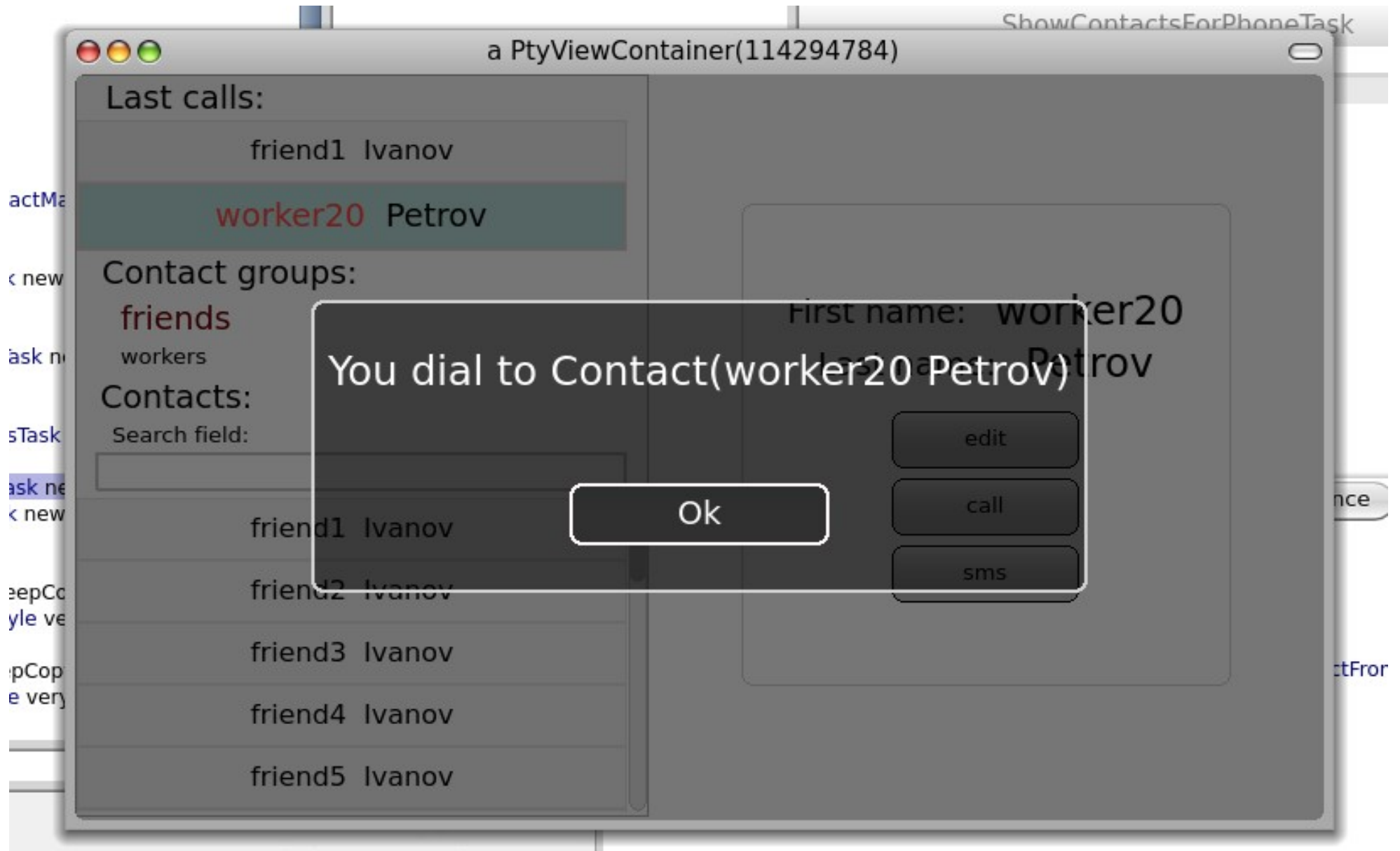
What should be done

- Configurable task activation
- Simple object editor task
 - user edit: object
- Better designer
- Serializing prototypes and settings to style classes
- Searching for prototypes and settings in multiple task contexts
- Documentation

Questions?

The end

Modal warning example



Tasks from blocks

DetailedContactPresenter>>runCall

```
guide executeTask: [
```

```
    guide doInModal: [
```

```
        guide user warn: 'You dial to ', model printString].
```

```
    guide user stopCurrentTask.
```

```
] asTask
```

Presenty UI framework

- Modified Model-View-Presenter pattern
- Continuation-based tasks as elements of user interaction at application level
- User requests as building blocks for tasks:
 - Primitive user requests to invoke base interface components
 - Compound requests for calling other tasks
- Extendable DSL (pure Smalltalk) facilitating task implementation
- Prototyping for visual components and tasks

Modified Model-View-Presenter pattern