The postMessage design outlined in the W3C document edited by Ian Hickson is not good!

The design of the cross document messaging by Ian Hickson (Google, Inc.) is very bad.
Even the last version is not good either. The design can be sketched here as follows.

```
The sender:
var o = document.getElementsByTagName('iframe')[0];
o.contentWindow.postMessage('Hello world', 'http://b.example.org/');


The receiver:
window.addEventListener('message', receiver, false);
function receiver(e) {
  if (e.origin == 'http://example.com') {
    if (e.data == 'Hello world') {
      e.source.postMessage('Hello', e.origin);
    } else {
      alert(e.data);
    }
  }
}
```

This design was messed up by pulling "origin" (a word that some people put too much attention more than should).
Even worse, it requires "o.contentWindow", this is really no professional sense.
Because of this design, if I open two tabs with the same url http://www.google.com/ they are not able to communicate.

My proposal is discard the "o.contentWindow" part requirement.

My better proposal

```
the sender:
window.postMessage(messageObject,targetDomain optional,windowIDs optional);

Either targetDomain or windowIDs should present.
I propose to use ID rather than name (though window can have a name),
since window.name is not required to be unique within the browser.


then the user agent(i.e. the browser, such as firefox) will do the following

var e={source: {href: get the sender's window.location.href,
```

```
                windowID: unique windowID within this browser
               },
       target: {domain: targetDomain as the sender requested,
                windows: the array of windowID
               },
       data: JSON.parse(JSON.stringify(messageObject)),
       ts: the timestamp when the post is requested
      };
if(windowIDs presents){
  postEvent to those windows.
} else {
  traverse the list of all windows
  for (each window){
    if(the domain of the window matches the target domain of the message) {
    postEvent(e);
  }
}


the receiver
/*
return true to indicate to continue to receive message from this sender
return false to indicate to deny messages from this sender forever
  (as long as the browser can remember this)
*/
function receiver(e) {
  if (e.source is accepted) {
    take the e.data to do whatever as desired.
    return true;
  }
  return false;
}

window.addEventListener('message', receiver, false);



if the receiver wants to respond to the sender
window.postMessage(messageObject,targetDomain optional,windowIDs optional);
        targetDomain can be found from e.source.href
        windowID can be found from e.source.windowID
        messageObject is the message object intended to be sent.
```

About domain match

the specification of the target domain can be

```
www.google.com
or google.com  this should match *.google.com
or com      this should match *.com
or ""  as for all
or https://www.google.com
```

```
or http://www.google.com:9876/abc/
```

For the last case, if a window.location.href=="http://www.google.com:9876/def/", then they do not match.

About Security

As long as the receiver check who is the sender which is identified by the user agent, there is no security issue at all.

About context sharing within the browser

Whether session data should be shared among the different processes of the same browser. such as cookies. It seems that firefox does not allow 2 different processes unless use different profile.

Here, one more setting, whether the windowIDs should be unique across different process. Within the same process among different tabs, they must be unique. If no more than one process is allowed, then such setting is not relevant.

Challenge

A bad design waste people's energy & time, to promote the better solution. I am offering a reward for the 1st one who implement my proposal. If you can do this before march 1st, 2013, I will give you $10.

jackiszhp@gmail.com

Last update: 2013.01.27 21:30