ACTA UNIVERSITATIS AGRICULTURAE ET SILVICULTURAE MENDELIANAE BRUNENSIS

Volume LX 35 Number 4, 2012

AUTOMATIC POLYGON LAYERS INTEGRATION AND ITS IMPLEMENTATION

O. Skoupý, D. Procházka

Received: January 24, 2012

Abstract

SKOUPÝ, O., PROCHÁZKA, D.: *Automatic polygon layers integration and its implementation.* Acta univ. agric. et silvic. Mendel. Brun., 2012, LX, No. 4, pp. 335–342

Land cover change analysis is one of the most important tools for landscape management purposes, as it enables exploring of long-term natural processes especially in contrast with anthropogenic factors. Such analysis is always dependent on quality of available data. Due to long tradition of map making and quality and accuracy of preserved historical cartographic data in the Czech Republic it is possible to perform an effective land use change analysis using maps dating even back to early nineteenth century. Clearly, because map making methodology has evolved since then, the primary problem of land cover change analysis are different sources and thus different formats of analyzed data which need to be integrated, both spatially and contextually, into one coherent data set. One of the most difficult problems is caused by the fact that due to different map acquisition methodologies the maps are loaded with various errors originating from measurement, map drawing, storage, digitalization and finally georeferencing and possible vectorization. This means that some apparent changes may be for example caused by different methodology and accuracy of mapping a landscape feature that has not actually changed its shape and spatial position through the time. This work deals with spatial integration of data, namely identifying corresponding lines in map layers from different epochs and adjusting the borders plotted in the less accurate map to spatially correspond to the more accurate map. For such a purpose, a special program had to be created. It basically follows the work by Malach *et al.*, 2009 who introduced their Layer Integrator. This work however presents a significantly different approach to creating an integration tool.

integration, sliver polygon, map layer, vectorization

Data integration

Attempts for integration of land use maps from different sources have been subject to several works. Their general goal is to increase confidence in change detection which can be achieved by many different methods. The most confident methods are still the manual ones which are often supported by the fact, that the created layers have to be vectorized manually anyway, so their vectorization can be performed under a set of rules that ensures spatial coherency of the whole data set as described by Skokanová, 2008 in the concept of backward vectorization and in Skoupý, 2008 by specifying a layer hierarchy according to spatial accuracy.

The other methods may be based on map generalization (Petit and Lambin 2002), which however negatively affects the accuracy of integrated data. Within the Czech Republic, geometrical land use maps integration has not been solved too frequently in land use change analysis applications (Malach *et al.*, 2009). In general, the issue of landscape data integration had been solved by Kolejka, 2002 and Kolejka, 2006 in his digital landscape model (DLM) concept. Automatic integration of different layers had been recently solved in work by Malach *et al.*, 2009 who introduced a tool named Layer Integrator. Even though the layer integration is a key issue in various GIS applications from flood protection (e.g. Machalová, 2009) to mapping the snow avalanche susceptibility (e.g. Suk *et al.*, 2011), many of the problems with data incoherence have still to be resolved.

MATERIALS AND METHODS

Cartographic data

The processed data comprise of two vector layers of Lower Morava Biosphere Reserve in Shapefile format, from mapping periods of about five years (completed in years 1990 and 2006). The data is the output of the research objective of the Ministry of Education, Youth and Sports MSM 6293359101 made by the Silva Tarouca Research Institute for Landscape and Ornamental Gardening. For its creation, Czechoslovak military topographic map from 1998–1995 at the scale of 1:25 000 and basic raster map of the Czech Republic from 2002–2006 at the scale of 1:10 000 were used. Both of the vector layers contain nine classes of land use. To gain a certain generalization corresponding with the 1:50 000 output scale, plots with areas below 0,8 ha were not vectorized. (Malach *et al.*, 2009).

Previously used methods

The goal of the work is to identify incorrectly plotted lines of the less precise (in this case older) land use map and replace them with identical lines from the more precise map. In the work of Malach *et al.*, 2009 which provided a basic inspiration, the layer integration process utilizes merging the two layers using a union tool and identifying sliver polygons, which supposedly do not represent a real temporal change in land use but they result from different planimetry of identical borders. As stated in the paper, these sliver polygons have following characteristics: Temporal change of land use (which is however not true in some special cases), narrow, elongated shape and small acreage.

The Layer Integrator, as designed by Malach *et al.* 2009 works exclusively with polygon vector layers. Two layers from different epochs were intersected by union tool. In the new layer, sliver polygons were identified by detecting a change of their land use code and their area to perimeter (A/P) ratio. Moreover, all the polygons with area equaling or lower then 0,5 ha were identified as sliver polygons. The idea of this approach was to dissolve the sliver polygons based on the land use code. All of the slivers change their land use code corresponding to the more precise (newer) layer. Then the dissolve tool is applied, merging the sliver polygon with its corresponding neighbor, thus eliminating the old, less precise border. This however works only if the land use codes of neighboring plots had not changed in between the two mapping epochs. If they have, the sliver polygon won't dissolve into any of the neighboring polygons and it will remain, having a different land use code from all of its neighboring polygons. Therefore, a corrective cycle had to be included, where all of these remaining sliver polygons are dissolved into a neighboring polygon that had changed its land use code between the epochs and was not marked as a sliver polygon.

This approach works in cases where up to one of neighboring polygons had changed its land use code. In fact, it fails in some rare cases where both of the neghboring plots change or just swap their land use code between the epochs. In the case of swapping, sliver polygons are not marked as sliver polygons because the change of land use code seemingly does not occur, so when dissolved, the more precise border is eliminated instead. In the case of land use change on both of the plots, there is no way to identify the less precise border to eliminate.

Results of the work by Malach *et al.*, 2009 shows that working solely with polygon layers and land use codes is not sufficient for creating a reliable layer integration tool.

Proposed solution

The solution is similar to that of the preceding work in some points. The area to perimeter (A/P) ratio has also been used for sliver polygon detection, but on the contrary, the rule of identifying sliver polygons by their temporal land use change was dropped. Moreover, one key characteristic was added. The new decision rule is based on the fact that for every sliver polygon the ratio between total lengths of borders originating from old and new layer has to be nearly equal. So unlike Malach *et al.*, 2009, who have employed solely polygon shapefile layers, the rewritten Layer Integrator also uses polyline shapefile layers to calculate the lengths of polygon borders from which a normalized length difference is calculated. Another advantage of polyline shapefile layers is getting a simplified topology information on neighboring polygons which appears to be crucial for proper and relatively quick tool functioning.

Also the rule of assigning a correct land use code to identified slivers had to be redefined. Newly, every sliver polygon (except for some special cases) is assigned a land use value of the neighboring polygon across the longest border originating from the old map layer. Some special cases occur when two sliver polygons would swap their land use code, in such a case the larger sliver polygon retains its original code value. This often results in preserving some smaller objects as lakes. Also, the code changes are performed in several iteration steps which assure that one sliver polygon does not receive a land use code from other sliver polygon that will yet have to change its own land use code (except for the aforementioned case), so the process has to begin with sliver polygons neighboring with non-sliver polygons. It should be noted that in this case we consider the union layer we are working with to be the final old land use map. The polygons can be thus consequently dissolved depending on their land use code from the older epoch, which eliminates all the lines which have been identified as incorrect and leaves us with a final corrected older land use map.

Implementation

As opposed to the work of Malach *et al.*, 2009, who have used an ArcGIS Model Builder for creating their tool, in our case we have decided for a hand written object-oriented Python script with ArcPy module which allows employing cursor tools for working with attribute tables of respective shapefile layers and thus enables constructing the tool according to our exact needs along with calling selected tools native to ArcGIS. The script is designed to be inserted in the command line in ArcGIS Python window, where the user may launch individual methods. The script has three methods that can be called by the user.

The initialization method asks for a directory path where the temporary data should be stored, names of newer and older (more and less precise) land use layers and names of land use code fields in their respective attribute tables. This method serves primarily to encapsulate the whole process in terms of object-oriented programming.

The first method being actually called by the user is the preparation method, which creates a union layer from designated shapefile layers, using the chosen directory to store the temporary files. The method contains a series of tools, beginning with the dissolve tool that eliminates possible borders between polygons bearing the same land use code. Attribute tables of the polygon layers are also complemented with an epoch field that identifies an epoch of the respective layer. In the next step, both polygon layers are merged by union tool and simultaneously converted into polyline format by "feature to line" tool so every line of the layer has an identifier which shows from which map the respective line was taken from. Since these tools are called by standard ArcPy commands, they are just verbally described and listed in an order in which they are carried out:

1. Dissolve the new land use map based on its land use codes.
2. Dissolve the old land use map based on its land use codes.
3. Copy land use codes of the old land use map into a newly created field.
4. Create a union layer of new and old land use maps ("Union.shp").
5. Use tool "Multipart to singlepart" to ensure every single polygon of the union layer represents one row in the layer's attribute table.
6. Create a polyline vector layer from the polygon union layer ("UnionPolylines.shp").
7. Add a new field into the new land use layer, identifying new boundaries and fill it with value "1" (attribute name is "Epoch1ID").
8. Add a new field into the old land use layer, identifying new boundaries and fill it with value "2" (attribute name is "Epoch2ID").
9. Create a polyline shapefile from both land use layers ("LandusesPolylines.shp").

In the next step, the polyline layer is treated with "delete identical" tool so the layer with an epoch indicator does not contain any spatially duplicitous records and every line has a unique number that identifies its spatial position. From the union layer, another polyline layer is created, this time for identifying adjacent polygons. The both polyline layers are merged together and lengths of the lines are calculated, so the resultant layer contains lines which identify the original layer they were taken from, spatial identifier, ID of adjacent polygons and length of the lines. Finally, the preparation method calculates area and perimeter for each polygon in the union layer, and totals lengths of bordering old and new lines, respectively. This process was implemented by following operations:

1. Delete all spatially identical objects in the layer "LandusesPolylines.shp".
2. Add a new field "Epoch" identifying old and new polygon borders. Then calculate the value of "Epoch" field by adding the values of field "Epoch1ID" to value of field "Epoch2ID". Number 1 thus represents a new border, value 2 represents an old border and value 3 represents a border that is spatially identical in both the old and the new land use map.
3. Add a field "SpatialCode" to "LandusesPolylines.shp" layer for a spatial identification of borders, fill it with FID+1 value to avoid zero values.
4. Spatially join both line layers "LandusesPolylines.shp" and "UnionPolylines.shp". ("Lines.shp" layer created). "Lines.shp" layer contains all objects of "UnionPolylines.shp" layer enhanced with spatial identification information, so all spatially identical lines bear the same "SpatialCode" number which further helps to search for the IDs of neighboring polygons and to transfer the land use codes between them.
5. Add a field "Length" to "Lines.shp" layer and fill it with calculated lengths of respective lines.
6. Add a field "Area" to "Union.shp" layer and fill it with calculated areas of respective polygons.
7. Add a field "Sliver" to "Union.shp" layer for sliver identification.
8. Add a field "NewPolBordLength" to "Union.shp" layer for totaling the length of new polygon borders.
9. Add a field "OldPolBordLength" to "Union.shp" layer for totaling the length of old polygon borders.
10. Add a field "SpatialCode" to "Union.shp" layer to identify a spatial code of respective transition border.
11. Add a field "SourceID" to "Union.shp" layer to identify a land use code source polygon for each sliver polygon.

Totaling of border lengths (“NewPolBordLength”, “OldPolBordLength”) depending on their origin (“Epoch” value):

```
unionRows = arcpy.UpdateCursor(“Union.shp”, “” , “” , “FID; NewPolBordLength; OldPolBordLength”, “”)
lineRows = arcpy.SearchCursor(“Lines.shp”, “” , “” , “Epoch; FID; Length; FID_Union”, “FID_Union A”)

lineRow = lineRows.next()

for unionRow in unionRows:
    lengthEpoch1ID=0.0
    lengthEpoch2ID=0.0

    while lineRow and lineRow.FID_Union <= unionRow.FID:
        if lineRow.Epo == 1 and lineRow.FID_Union == unionRow.FID:
            lengthEpoch1ID = lengthEpoch1ID + lineRow.Length
        elif lineRow.Epo == 2 and lineRow.FID_Union == unionRow.FID:
            lengthEpoch2ID = lengthEpoch2ID + lineRow.Length
        if lineRow:
            lineRow = lineRows.next()

unionRow.NewPolBordLength = lengthEpoch1ID
unionRow.OldPolBordLength = lengthEpoch2ID
unionRows.updateRow(unionRow)
```

The second method the user can call is a sliver decision tool that identifies slivers depending on chosen characteristics, i.e. the minimal area, area to perimeter (A/P) ratio and normalized difference between lengths of borders originating from the old and new layer. The method calculates these values for every polygon and immediately decides whether the polygon meets the chosen criteria or not. The sliver identification is implemented by following algorithm:

```
unionRows = arcpy.UpdateCursor(“Union.shp”, “” , “” , “Sliver; Area; Perimeter; NewPolBordLength; OldPolBordLength”, “”)

for unionRow in unionRows:
    if (unionRow.Area/unionRow.Perimeter) < minAPRatio or unionRow.Area < minArea:
        if (abs(unionRow.NewPolBordLength-unionRow.OldPolBordLength))/(unionRow.perimeter) < minLRatio:
            unionRow.Sliver = 1
        else:
            unionRow.Sliver = 0
    else:
        unionRow.Sliver = 0
    unionRows.updateRow(unionRow)
```

Consequently, user can decide whether he wants to further process the layer, in such a case he can call the third method, which creates a final land use map and saves it under a selected file name. In the first step, every sliver polygon is assigned a spatial code value of the longest bordering line originating from the older layer. If any two polygons could be assigned an identical spatial code value, then the larger of the two is not considered to be a sliver polygon anymore. For each sliver, identify the longest old border for code transformation:

```
unionRows = arcpy.UpdateCursor(“Union.shp”, “Sliver = 1” , “” , “SpatialCode; FID” , “”)
lineRows = arcpy.SearchCursor(“Lines.shp”, “Epoch = 2” , “” , “Length; FID_Union; SpatialCode”, “FID_Union A; Length D”)

lineRow = lineRows.next()

for unionRow in unionRows:
    while lineRow and lineRow.FID_Union < unionRow.FID:
        lineRow = lineRows.next()
    unionRow.SpatialCode = lineRow.SpatialCode
    unionRows.updateRow(unionRow)
    lineRow = lineRows.next()
```

In the next step, every sliver polygon is assigned an FID of adjacent source polygon that provides its correct land use code to the sliver. For each sliver, record an FID of source polygon, omit larger slivers in case of possible mutual swap of land use codes. Assign a “Sliver” field value of “2” to such unused slivers. Assign a “Sliver” field value of “3” to slivers that cannot receive the land use code:

```
unionRows = arcpy.UpdateCursor(“Union.shp”, “Sliver = 1 or Sliver = 2 or Sliver = 3” , “” , “SpatialCode; SourceID; Area”, “SpatialCode D, Area A”)
lineRows = arcpy.UpdateCursor(“Lines.shp”, “Epoch = 2” , “” , “FID_Union; SpatialCode”, “SpatialCode D”)
lineRow = lineRows.next()

for unionRow in unionRows:

    while lineRow and unionRow.SpatialCode < lineRow.SpatialCode:
        lineRow = lineRows.next()
    if lineRow:
        if unionRow.SpatialCode == lineRow.SpatialCode:
            if unionRow.FID == lineRow.FID_Union:
                lineRow = lineRows.next()
                if lineRow and unionRow.SpatialCode == lineRow.SpatialCode:
                    unionRow.SourceID = lineRow.FID_Union + 1
                    unionRow.sliver = 1
                    unionRows.updateRow(unionRow)
                else:
                    unionRow.sliver = 3
                    unionRows.updateRow(unionRow)
            else:
                unionRow.SourceID = lineRow.FID_Union + 1
                unionRow.sliver = 1
            unionRows.updateRow(unionRow)
            lineRow = lineRows.next()
        while lineRow and unionRow.SpatialCode == lineRow.SpatialCode:
            lineRow = lineRows.next()
    else:
        unionRow.sliver = 2
        unionRows.updateRow(unionRow)
```

After this step, the iteration process is launched, during which non-sliver source polygons transmit their land use code to respective adjacent sliver polygons. The cycle assigns correct old land use values (temporarily stored in the "OLField" attribute table field) to sliver polygons. All non-sliver polygons that are considered a source will give their land use code to respective adjacent sliver polygon. During the code change, the Sliver polygon changes its state to "Treated sliver" ("Sliver" = 4) and may act as a source during the next iteration. The "counter" variable registers number of performed iterations, the "useCheck" variable ensures the cycle is terminated if no more changes in land use codes are performed:

```
counter = 0
useCheck = 1

while useCheck == 1:

    useCheck = 0
    sliverRows = arcpy.UpdateCursor("Union.shp", "Sliver=1" , "" , "FID; OLField; SourceID; Sliver" , "SourceID A")
    sourceRows = arcpy.SearchCursor("Union.shp", "Sliver=0 or Sliver=2 or Sliver=4" , "" , "OLField", "")

    sliverRow = sliverRows.next()

    for sourceRow in sourceRows:
        while sliverRow and (sliverRow.SourceID - 1) < sourceRow.FID:
            sliverRow = sliverRows.next()
        while sliverRow and (sliverRow.SourceID - 1) == sourceRow.FID:
            sliverRow.OLField = sourceRow.OLField
            sliverRow.Sliver = 4
            sliverRows.updateRow(sliverRow)
            sliverRow = sliverRows.next()
            useCheck = 1
        if sliverRow:
            print sliverRow.SourceID
    counter = counter + 1
```

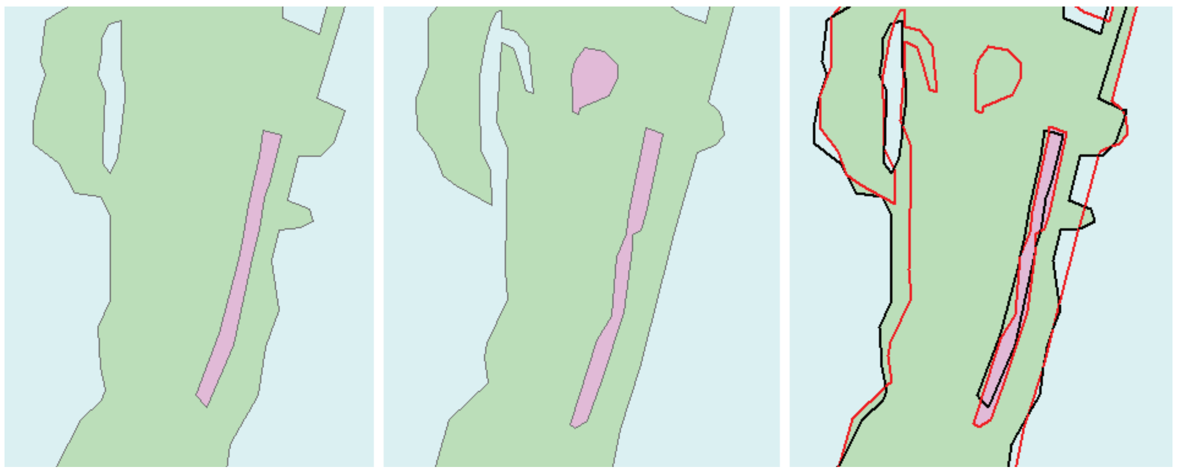
As shown, treated sliver polygons are immediately marked as non-sliver polygons and after finding all possible changes, the next iteration is launched. In the end, the layer is dissolved on a basis of the old land use field that creates the adjusted old land use map:

Save the updated land use codes "OLField" of the "Union.shp" layer in original user defined field (subtract "1" to return to the original land use code numbering).

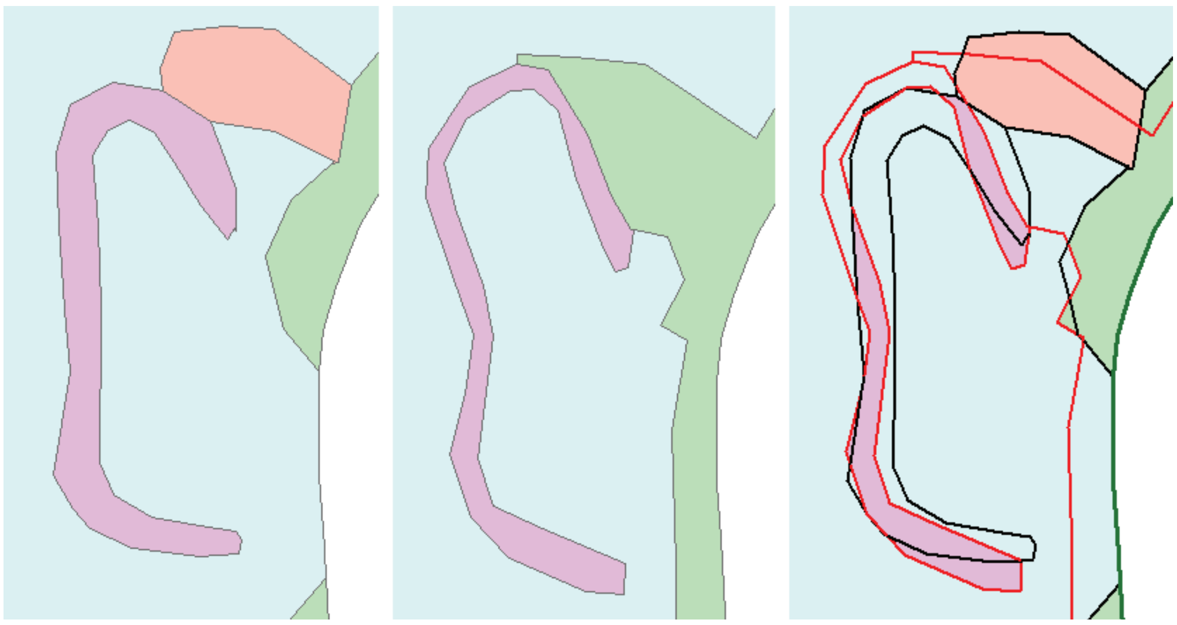
Create the output layer by dissolving the "Union.shp" file based on the Old land use code field values. (User defined output created).

RESULTS AND DISCUSSION

The complete process of integrating the new and old layer with ca. 1 190 and 920 polygons respectively has lasted 58 minutes (using 3.21 GHz quad-core processor), as the number of processed polygons totaled at 8 375. However, there occurred a removal of some smaller objects from the older land use map, as they have been divided into two or more polygons that have been identified as sliver polygons and have been dissolved into neighboring polygons. On the other hand, there did not occur any significant removal of elongated polygons that have been often eliminated by version provided by Malach *et al.*, 2009 that did not take different origin of polygon borders into account. An example of layer integration can be seen in Fig. 1.



1: *An example of correct layer integration. 1990 layer (left), 2006 layer (middle) and corrected 1990 layer (right) with highlighted lines originating both from the older (black) and the newer (red) layer.*



2: *An example of incorrectly corrected boundaries due to low precision and unsuitable settings of the decision criteria. 1990 layer (left), 2006 layer (middle) and corrected 1990 layer (right) with highlighted lines originating both from the older (black) the newer (red) and both (green) layer. As we can see, a large portion of elongated lake polygon (purple) in the 1990 layer has been marked as a sliver polygon and dissolved into a neighboring (cyan) polygon. Additionally, the upper left part of the polygon of 2006 layer which should compensate the omission and preserve the lake polygon in its full length, was not identified as a sliver and was not assigned a lake land use code, due to disproportion between borders originating from the old and the new layer.*

It should be noted that the results are actually dependent on the layers accuracy and user settings. The use of Layer integrator should primarily be recommended to slightly adjust spatially accurate data, in case of higher inaccuracies, a number of incorrectly identified polygons inevitably grows, one of such cases can be seen in Fig. 2.

Maybe, another parameters could be added, enabling to set a maximal possible area of a sliver polygon, which would enable to avoid some of the errors as described above. Furthermore, another addition could comprise from the possibility to choose whether to take a difference of old and new land use into account, as suggested in Malach *et al.*, 2009. This aspect is however not absolutely reliable since an unchanging land use of a polygon does not automatically guarantee non-sliverness. But it can help to preserve some existing elongated polygons in the integrated layer at least to some extent, preventing their complete disappearance.

CONCLUSION

The results proved to be very satisfactory, although the layer integration itself is highly dependent on the user settings. In either way, the resulting, completely rewritten script means a significant improvement over its predecessor, the original Layer Integrator presented in Malach *et al.*, 2009 which was constructed solely within the ArcGIS Model Builder. This new tool provides a more compact and flexible solution which may be useful in various applications related to land cover change analysis such as Zidek & Klimanek, 2010.

In general, the benefit of automatic integration is in a consistent approach to integration as opposed to manual methods, whilst one of the main setbacks is an inability to recognize sliver polygons in some distinct special cases, mostly when slivers form a part of larger polygons, which can lead to partially incorrect results. In any case, the automatic integration with some degree of supervision is surely a more effective method to integrate already vectorized layers, whereas in the process of vectorizing a new data the manual integration should be recommended, as it increasingly reduces a need to vectorize a larger number of lines from the older (or the more accurate) layers.

SUMMARY

The aim of the work was to create a polygon layers integration tool for creating spatially coherent data sets from map layers with different spatial accuracy. This can be done by adjusting borders of the less precise maps to fit identical lines of the more precise maps. The solution was performed through developing an object-oriented Python script using ArcPy module. The tool works with two polygon shapefile layers, in this case maps of land use on the area of Lower Morava Biosphere Reserve in

different years (1990 and 2006). The principle of the borders correction is based on identifying sliver polygons in a union layer. Sliver polygons appear as a result of inevitable differences in planimetry that are caused by various errors during the process of map creation. The criteria for identifying sliver polygon are area to perimeter (A/P) ratio, area of the polygon and normalized difference between totaled lengths of polygon borders originating from the older and the newer map. Identified sliver polygons are then assigned a correct land use code. Consequently, they are dissolved into neighboring polygons thus eliminating the incorrectly drawn borders and preserving the more precise borders from the newer map layer, creating a corrected land use map of the older epoch. The tool can be launched by inserting the script in the command line in ArcGIS Python window and it is designed to be controlled by a set of commands specifying a method to be launched along with parameters for sliver identification.
The efficiency of layer integration proved to be highly dependent on user settings and spatial accuracy of used layers. The identification of true sliver polygons is more reliable with a set of more accurate layers as opposed to inaccurate layers where looser decision rules have to be applied. The area of the layer and map scale are also a factor when considering the tool efficiency, since they affect the computing time. In our case where two layers with 1 190 and 920 polygons were merged together thus creating a union layer of 8 375 polygons the complete computing process took 58 minutes with 3.21 GHz quad-core processor. Given the moderate precision of the used data, the tool managed to identify most of the slivers and integrate most of the presumably identical borders whilst preserving most of smaller polygons such as lakes and rivers which could be mistaken for sliver polygons.

REFERENCES

KOLEJKA, J., 2002: Digital landscape model as integrated database tool. In: Ruzicka, J. (ed.): GIS Ostrava. Technical University Ostrava.

KOLEJKA, J., 2006: Digital landscape model and its utilization in primary and applied geographic research. *Geographia Technica*, Vol. 1, No. 1, s. 103–110.

MACHALOVÁ, J., 2009: Space modeling in management of landscape flood-protection measures. *Acta Universitatis agriculturae et silviculturae Mendelianae Brunensis,* sv. LVII, No. 6, s. 133–142. ISSN 1211-8516.

MALACH, S., KLIMÁNEK, M., DOUDA, P., SKOKANOVÁ, H., STRÁNSKÁ, T., 2009: Spatial data integration for land use change analysis of the Lower Morava Biosphere Reserve, *Geoscape*, Vol. 4, No. 1, s. 100–112.

PETIT, C. C., LAMBIN, E. F., 2002: Impact of data integration technique on historical land-use/land cover change: Comparing historical maps with remote sensing data in the Belgian Ardennes. *Landscape ecology*, Vol. 17, No. 2, s. 117–132.

SKOKANOVÁ, H., 2008: GIS methods in land use changes assesments. In: *Geoinformatics in public administration*. Czech association for Geoinformation.

SKOUPÝ, O., 2008: Analýza změn krajinného pokryvu. Brno, 43 s. Diplomová práce na Stavební fakultě Vysokého učení technického na Ústavu geodézie. Vedoucí diplomové práce Ladislav Plánka.

SUK, P., KLIMÁNEK, M., 2011: Creation of the snow avalanche susceptibility map of the Krkonoše Mountains using GIS. *Acta Universitatis agriculturae et silviculturae Mendelianae Brunensis,* sv. 59, č. 5, s. 237–246. ISSN 1211-8516.

ŽIDEK, V., KLIMÁNEK, M., 2010: Applied Geoinformatics in Forestry and Landscape Research and Education. In: Soomro, S. New Achievements in Technology, Education and Development. 1. ed. Vienna: In-Tech, s. 151–176. ISBN 978-953-307-066-7.

Address

Ing. Ondřej Skoupý, Ústav geoinformačních technologií, Lesnická a dřevařská fakulta, Mendelova univerzita v Brně, Zemědělská 1, 613 000 Brno, Česká republika, Ing. David Procházka, Ph.D., Ústav informatiky, Provozně ekonomická fakulta, Mendelova univerzita v Brně, Zamědělská 1, 613 000 Brno, Česká republika, e-mail: xskoupy1@node.mendelu.cz, david.prochazka@mendelu.cz